GESELLSCHAFT
FÜR INFORMATIK

Fachausschuss

Management der Anwendungsentwicklung und -wartung (WI-MAW)

im FB Wirtschaftsinformatik

Jahrgang 28 Heft 1
ISSN 1610-5753

März 2022

Inhalt

46. WI-MAW-Rundbrief

Inhaltsverzeichnis

Fachbeiträge

Ankündigungen

Berichte

Organisation

DevOps in Zeiten von Regulierung und Zero Trust

Jens Borchers

Beratung für Informationsmanagement / cat out GmbH

jens@borchers-bfi.de / jens.borchers@cat-out.com

Abstract: Man muss kein IT-Sicherheitsexperte sein, um jeden Tag von neuen Angriffen und Unternehmen und andere Organisation zu erfahren, bei denen entweder in großem Stil Daten gestohlen („Data Leakage“) oder – für die Unternehmen noch schlimmer – ganze Datenbestände von Externen so verschlüsselt werden („Ransomware“), dass sie in der Regel für das Unternehmen verloren sind, wenn das geforderte Lösegeld nicht gezahlt wird, um die Schlüssel zu erhalten. In den letzten Jahren hat sich die IT-Welt stark in Richtung Cloud-basierter Systeme verschoben, die eine Eingrenzung der „inneren IT“ immer schwieriger macht. Auch die Entwicklungsmethoden und Tool-Ketten für eine schnellere „Time-to-market“ haben sich in den letzten Jahren unter dem Begriff „DevOps“ stark verbreitet, und dieses vor allem im Cloud-Bereich. Vor diesem Hintergrund sind die altbekannten Absicherungsmechanismen wie Firewalls, DDoS-Abwehr etc. allein nicht mehr ausreichend. „Zero Trust“-Ansätze sollen dafür sorgen, sicherheits-technisch flexibler auf jeden einzelnen Benutzer-Request reagieren zu können, wobei unterstellt wird, dass zunächst jeder verdächtig ist: „Traue niemandem, überprüfe jeden!“ ist die Kultur der IT-Sicherheits-Zukunft, auch im Bereich der Entwicklungssysteme.

Schlüsselbegriffe: DevOps, DeveSecOps, Cybersecurity, Zero Trust, Sicherheitsarchitekturen

Dieser Beitrag ist wie folgt aufgebaut:

1. Einführung und Rückblick
2. Überblick über regulatorische Anforderungen
3. DevOps und Sicherheitsanforderungen
4. Zero Trust als aktuelle Sicherheitsarchitektur
5. DevSecOps als Umsetzung von Zero Trust

1 Einführung und Rückblick

Es bedarf eigentlich keiner weiteren großen Motivation, das Thema „Cybersicherheit“ auch im Bereich der Entwicklung, des Testens und des Deployments von Software stärker in den Fokus zu nehmen, soweit es nicht schon geschehen ist. Ein ausführlicher Überblick über die aktuelle Sicherheitslage in Deutschland findet sich u.a. in [BSI 2021]. Die vielfältigen Sicherheitsvorfälle, auch im Hinblick auf Nicht-Produktionsumgebungen, genutzte Open-Source-Software oder gar Entwicklungs- oder andere Werkzeuge, geben besonderen Anlass zur Aufmerksamkeit. Gerade im Zeitalter agiler Entwicklung und steigender Anforderungen an die „time-to-market“ haben sich auch entsprechende DevOps-Konzepte mehr und mehr verbreitet.

Hatten im letzten DevOps-Fachbeitrag des Autors in dieser Reihe [BORCH 2018] noch die rechtlichen und regulatorischen Anforderungen von DevOps-Konzepten gerade im Finanzbereich im Vordergrund gestanden, so hat sich die Aufmerksamkeit heute auch auf die Sicherheit entsprechender Mechanismen verlagert.

Leider wird in vielen Unternehmen noch immer zu wenig Gewicht darauf gelegt, ihren Entwicklern zu verdeutlichen, dass auch ihre Prozesse und die Qualität in der Entwicklung wesentlich zur Gesamtsicherheit eines IT-Betriebs beitragen können und müssen.

2 Überblick über regulatorische Anforderungen

Der Betrieb – und dazu gehört im weiteren Sinne auch die Entwicklung und Wartung aller Anwendungssysteme - von IT-Umgebungen unterliegt heute einer Vielzahl von internationalen und nationalen Gesetzen, Verordnungen, Standards, die hier nicht alle behandelt werden können (eine detailliertere Übersicht enthält [BORCH 2019]).

Eine grobe exemplarische Übersicht über das gesetzliche „Spinnennetz“ bezüglich der IT-Sicherheit und Datenschutz z.B. einer großen internationalen Versicherung hat z.B. [SIMON 2020] aufgezeigt:

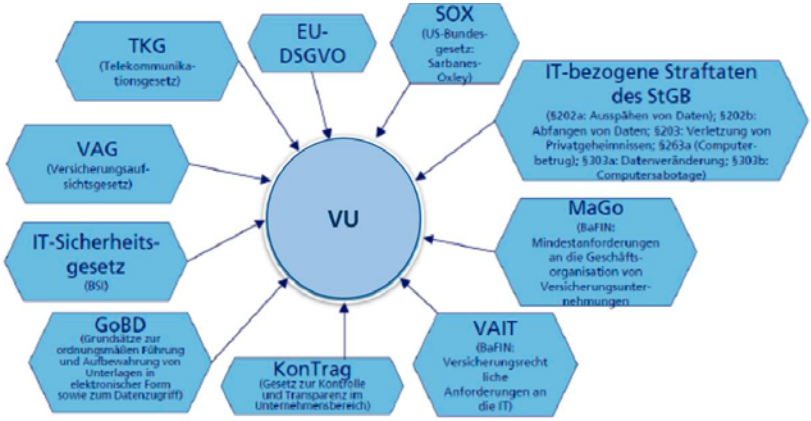


Abbildung 1: Gesetzliche IT-Rahmenbedingungen einer Versicherung
(nach [SIMON 2020])

Die wesentlichen rechtlichen Rahmenbedingungen können wie folgt zusammengefasst werden:

- Die EU-Datenschutzgrundverordnung der EU (EU-DSGVO) und ihre Umsetzung in deutsches Bundes-Datenschutzrecht haben per 25.05.2018 zu einem massiven Anpassungsdruck für die Entwicklung und vor allem den Betrieb von allen IT-Systemen geführt, in denen Daten natürlicher Personen gespeichert und verarbeitet werden. Insbesondere aus dem Artikel 32 („Sicherheit der Verarbeitung“) ergeben sich zahlreiche Anforderungen an die technischen und organisatorischen Maßnahmen zum Betrieb der entsprechenden Systeme. Dieses gilt in extremen Maß, wenn auch personenbezogene Daten über Webseiten oder APIs nach außen zugänglich sind. Die bereits bekannten Verstöße seit 2018 zeigen, dass die Strafen durchaus drakonisch sein können und mit dem „zahnlosen Tiger“ des alten BDSG (bei dem die Strafen zumeist im niedrigen vierstelligen EUR-Bereich lagen) nichts mehr gemein haben. Darüber hinaus kommen die Schadenersatzansprüche betroffener Personen und natürlich in jedem Fall der Image-Schaden hinzu.
- Neben der EU-DSGVO zunächst etwas aus dem Fokus geraten, obwohl fast zum selben Zeitpunkt (im Juni 2018) aktiv geworden, ist die EU-Richtlinie „über den Schutz vertraulichen Know-hows und vertraulicher Geschäftsinformationen (Geschäftsgeheimnisse) vor rechtswidrigem Erwerb sowie rechtswidriger Nutzung und Offenlegung“, die in Deutschland erst am 18.04.2019 durch das „Gesetz zum Schutz von Geschäftsgeheimnissen (GeschGehG)“ in geltendes Recht eingegangen ist. Dieses Gesetz fordert von den Unternehmen eine umfassende Absicherung ihrer kritischen Assets, d.h. insbesondere aller Dokumente,

die für das Überleben des Unternehmens unabdingbar sind. Das Gesetz fordert dabei nicht nur IT-technische, sondern auch entsprechende organisatorische Absicherungen bis hin zur umfassenden Sicherheits-Bewertung von Bewerbern und der Ausgestaltung von Arbeitsverträgen. Auch hier spiegelt sich bereits eine – nicht nur technische – „Zero Trust"-Kultur wider, da u.a. unterstellt wird, dass Bedrohungen nicht nur von außen, sondern durchaus auch von Mitarbeitern (besonders ehemaligen) und anderen Dienstleistern im eigenen Unternehmen ausgehen können. In [Bitkom 2018] wird dieses deutlich nachgewiesen:

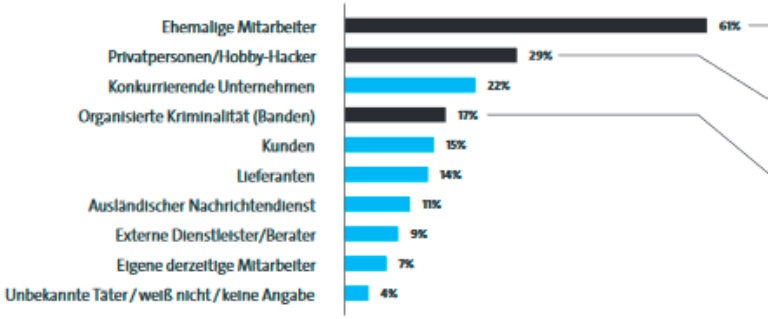


Abbildung 2: Täterkreis bei betroffenen Industrieunternehmen [BITKOM 2018]

- Während die beiden obigen Gesetzeskomplexe für alle Unternehmen gleichermaßen gelten, insbesondere bei der Speicherung personenbezogener Daten (und das gilt schon allein durch die Beschäftigtendaten de facto für alle Unternehmen), ist mit dem IT-Sicherheitsgesetz durch das „Bundesamt für Sicherheit in der Informationsverarbeitung" (BSI) bereits seit 2009 („BSI-Gesetz") und der Erweiterung 2015 („KRITIS") ein Katalog mit hohen Sicherheitsanforderungen an Unternehmen, die zu den kritischen Infrastrukturen gehören, aufgestellt worden. Mit dem im Mai 2021 weiter verschärften „IT-Sicherheitsgesetz 2.0" wird sowohl der Kreis der betroffenen Unternehmen als auch deren Verantwortung und Berichterstattung gegenüber dem BSI massiv erweitert.
- Dabei zählen wesentlich mehr Unternehmen zur „kritischen Infrastruktur", als man landläufig annehmen könnte, also nicht nur Netzversorger im Telekommunikations-, Energie- und Verkehrsbereich sowie Krankenhäuser und andere öffentliche Einrichtungen. Auch private Unternehmen wie zentrale Anbieter bundesweiter Services z.B. im Finanzdienstleistungsumfeld gehören zu den von KRITIS betroffenen Unternehmen. Die Umsetzung ist auf Basis der vom BSI herausgegebenen Standardhandbüchern 200-1, 200-2, 200-3 und des aktuellen IT-Grundschutz-Kompendiums nachzuweisen und zu zertifizieren.
- Neben den drei oben aufgeführten wesentlichen Gesetzen existieren diverse weitere Vorgaben entweder durch den Gesetzgeber oder Standardisierungsgremien der Unternehmen, die weitgehende IT-Sicherheitsmaßnahmen fordern. Dazu gehören z.B. der Finanzbereich mit den von der BaFin geforderten Risikomanagement-Maßnahmen BAIT, VAIT, KAIT ebenso wie die Automobil- und andere Industrien mit sicherheitskritischen Produkten.
- Für viele Unternehmen ist es heute unausweichlich, durch Erlangung bestimmter Zertifikate nachzuweisen, dass sie die geforderten Anforderungen an die IT-Sicherheit umgesetzt haben. Die bekannteste Zertifizierung neben dem vom BSI geforderten IT-Grundschutzzertifikat (sofern man diesem Gesetz unterliegt) ist sicherlich die auf Basis der ISO/IEC-Norm 27001 (und zugehörige).

3 DevOps-Konzepte und Sicherheitsanforderungen

Dieser Beitrag hat nicht das Ziel, die Ziele und Ansätze agiler Software-Entwicklung und der später ergänzten DevOps-Konzepte im Einzelnen darzustellen. Dazu wird auf die mittlerweile umfangreiche Literatur zu diesen Themenkreisen verwiesen. Hier sollen aus Sicht des Autors nur die wichtigsten Konzepte, insbesondere im Hinblick auf die mögliche Konfrontation mit der Regulierung behandelt werden.

3.1 Agile Software-Entwicklung

Agile Konzepte haben sich seit dem im Jahr 2001 veröffentlichten „Manifesto for Agile Software Development", welches seinerzeit von 17 führenden Software-„Gurus" unterzeichnet wurde, immer weiter durchgesetzt, wobei sich vor allem die Projektmanagementmethode „Scrum" als wesentlicher Ansatz etabliert hat. Häufig wird Scrum aber fälschlich als Vorgehensmodell zur Herstellung von Software interpretiert, der Unterschied zu z.B. „Extreme Programming" als einer möglichen agilen Entwicklungsmethodik wird nicht immer klar.

Eigentlicher Nutznießer der agilen Konzepte sollen die Anforderer aus den Fachbereichen (in Scrum-Projekten vertreten durch den „Product Owner") sein, die sehr schnell auf schon umgesetzte Funktionen reagieren können und nicht mehr gezwungen sind, ein vollständiges Fachkonzept zu liefern, bevor die eigentliche Entwicklung beginnt.

3.2 DevOps-Konzepte

Parallel zur starken Verbreitung von agilen Entwicklungskonzepten werden bereits in vielen Unternehmen auch DevOps-Konzepte verfolgt. Diese sind auch bereits in [BORCH 2018] dargestellt.

Um den mit den agilen Entwicklungsansätzen verbundenen Anspruch an die Änderungsgeschwindigkeit nicht an der Hürde „Übergang in die Produktion" enden zu lassen, haben

Patrick Debois und Andrew Shafer erstmals 2007 (siehe dazu [PAUL 2014]) einen dynamischen Übergang von Software aus der Entwicklung in die Produktion propagiert. Dieser wurde 2009 durch John Allspaw und Paul Hammon in "10+ Deploys a Day: Dev and Ops Cooperation at Flickr" [ALLSHAM 2009] bekannt gemacht.

Allerdings bleibt in vielen Unternehmen trotzdem eine mehr oder weniger strikte organisatorische und technische Trennung von Entwicklung und Wartung („Change the Business") einerseits und dem IT-Produktionsbetrieb („Run the Business") bestehen. Diese Trennung basiert primär auf dem bekannten ITIL (IT Infrastructure Library)- Framework, das seit 2005 als „ISO 20000" auch als Standard für Organisationen zur Verfügung steht. ITIL bildet seitdem auch die Vorgabe zur Umsetzung der o.g. regulatorischer Vorgaben für den Übergang von Software-Artefakten aus der Entwicklungs- in die Produktionsdomäne.

Das Hauptziel von DevOps ist es eigentlich, die durch das ITIL-Modell verstärkte „mentale Trennung" von Entwicklung und Betrieb wieder zu reduzieren. Während die Entwickler alle ihre Neuerungen so schnell wie möglich auch in Betrieb nehmen wollen, geht es diesem – nicht ganz zu Unrecht – im Wesentlichen um einen stabilen und reibungslosen Ablauf der IT-Produktion. Auf der anderen Seite werden zum Teil immer noch – auch in agilen Projekten – die späteren Betriebsaspekte viel zu spät mit in die Entwicklungsvorgaben aufgenommen, was dann spätestens beim ersten Release zu unnötigen Diskussionen und Nacharbeiten führt.

In vielen Unternehmen wird DevOps nur mit dem Anspruch eingeführt, im Sinne eines „Continuous Integration"/„Continuous Deployments" (CI/CD) die technische (Zeit-)Strecke zwi-

schen dem Freigeben einer Änderung durch den Entwickler bis zur Aktivierung in der IT-Produktion so zu optimieren, dass diese einer minimalen Zeit erfolgen kann. Dieses wird durch entsprechenden Tooleinsatz realisiert, der häufig auf dem Kubernetes-Produkt basiert.

Was allerdings für ein Start-up-Unternehmen in einem nicht regulierten Bereich an DevOps-Konzepten direkt einsetzbar sein mag, ist z.B. bei einem Finanzdienstleister aufgrund der regulatorischen Anforderungen nur bedingt und auch nur mit erweiterten Rahmenbedingungen möglich (siehe dazu [BORCH 2018]).

3.3 DevOps- und Entwicklung-Sicherheitsanforderungen

Die Bedrohungen durch externe Angreifer nehmen kontinuierlich zu und sie beschränken sich schon lange nicht mehr direkt auf die Produktionssysteme, sondern suchen sich andere „Türöffner“ auch im Entwicklungs- und Testumfeld, das bei vielen Unternehmen immer noch bessere Angriffsmöglichkeiten bietet.

Durch das von DevOps wieder verstärkt geförderte „Zusammenrücken“ von Entwicklung und Produktion ergeben sich Angriffsmöglichkeiten, die bei der starren Trennung (nach ITIL) so nicht gegeben waren. Andererseits sind andere Angriffsvektoren unabhängig davon, ob DevOps eingesetzt wird oder nicht.

Die primären Angriffsmöglichkeiten lassen sich folgenden groben Kategorien zuordnen:

- Authentifizierung und Autorisierung

  Der Klassiker unter den Bedrohungen. Ohne hier alle Aspekte von Angriffen auf User-Accounts darstellen zu wollen, ist dieses immer noch der größte Angriffsvektor. Dazu zählen folgende Angriffsziele:

  - Erbeutung von Zugangsdaten durch Phishing oder auch Attacken ähnlicher Art bis hin zum „Social Engineering“. Dazu ist gerade in den letzten beiden Jahren bis hin in die Tagespresse umfassend berichtet und gewarnt worden, aber nicht immer bringen Benutzer trotz umfassender Trainings dazu die nötige Aufmerksamkeit und Disziplin auf, z.B. verdächtige Mails etc. direkt zu löschen oder anderen „schnell zu helfen“, wenn diese angeblich ein Zugangsproblem haben und unter Zeitdruck stehen.

  - Dieses wird verstärkt, wenn Benutzer aus dem Entwicklungsumfeld mit identischen Zugangsdaten auch Zugriff auf Produktivsysteme haben oder Single-Signon-Systeme eingesetzt werden, bei denen mit einem Login automatisch diverse Systeme erreicht werden, ohne dass es einer erneuten Anmeldung bedarf.

  - Nicht vergessen werden dürfen dabei sogenannte „privilegierte Benutzer“ wie Administratoren, speziell z.B. die für Deployment-Prozesse eingerichteten technischen Benutzer, die keiner natürlichen Person zugeordnet sind, aber über wesentliche Rechte auf beiden Umgebungen verfügen

  - Wenn für Produktionsumgebungen keine zusätzlichen Absicherungen eingesetzt werden wie z.B. Multi-Faktor-Authentifizierung, ergibt sich über Entwicklungsumgebungen eine unerwünschte Zugangsmöglichkeit.

  - Last but not least bleibt leider immer noch die Möglichkeit, dass sich Hacker in dem später anzugreifenden Unternehmen anstellen lassen, möglichst auf Positionen mit vielen Rechten wie z.B. Administratoren. Dieses ist bei großen „lohnenden Zielen“ eine nicht auszuschließende Option. Dazu stehen mittlerweile Checklisten auch vom BKA [BKA2017] für privatwirtschaftliche Unternehmen zur Verfügung, die früher nur im Hochsicherheits- und militärischem Bereich üblich waren.

- Die Infrastruktur zur Verbindung von Entwicklung und Produktion

  Eine fehlende topologische Trennung von Produktions- und Nicht-Produktionsumgebungen, also Entwicklung, Test, Abnahme, liefert Angreifern ein Angriffsziel, wenn sie es schaffen, sich über Entwicklungssysteme auch zu Produktionssystemen zu bewegen. Dieses hat häufig folgende Gründe:

  - Unzureichende Servertrennung
    Leider findet sich in diversen Rechenzentren immer noch keine strikte physische Trennung von Entwicklungs- und Produktions-Umgebungen. So sind zwar häufig getrennte virtuelle Maschinen vorhanden, aber diese sind oft nicht sauber in unterschiedlichen Clustern zugeordnet und/oder die virtuellen Maschinen residieren zusammen auf einer physischen Hardware-Plattform.

  - Fehlende Netzwerkabsicherung
    Die Netzsegmente für Entwicklung und Produktion sind nicht maximal topologisch getrennt. Gerade wenn DevOps eingesetzt werden soll, bestehen häufig aber derartige (offene) Verbindungen, um die automatisierten Deployments einfacher zu machen.

- Der DevOps-Prozess und die zughörige Tool-Kette

  Heute wird der Deployment-Prozess durch hoch-automatisierte Tool-Ketten abgebildet. Auch diese können von Angreifer genutzt werden.

  - Einschleusen von Komponenten

    Automatisierte DeveOps-Prozesse können genutzt werden, um über die Entwicklungsumgebung eigene Software in die Produktion einzuschleusen oder Komponenten durch entsprechend manipulierte zu ersetzen.

  - Manipulieren von Tool- und Runtime-Parametern

    Neben dem Einschleusen von ganzen Komponenten kann auch die Manipulation einzelner Parameter zu einem Sicherheitsrisiko für den Produktionsbetrieb führen.

  - Manipulieren von Entwicklungs-Tools

    Spätestens seit dem Bekanntwerden des „SolarWinds“-Vorfall kann nicht mehr ausgeschlossen werden, dass Angreifer auch Entwicklungs- oder andere Werkzeuge manipulieren, um diese dann als Angriffswaffe gegen das eigentliche Ziel einzusetzen. Da viele Entwicklungs-Tools aus dem Open-Source-Umfeld stammen, ist hier die Gefahr besonders groß.

- Die Software-Komponenten selbst

  - Eigene – unbemerkt manipulierte – Software

    Bei Eigenentwicklungen kann bei Verzicht auf statische Code-Prüfungen auch in Bezug auf Sicherheitsaspekte (die ja von MITRE.ORG immer aktuell publiziert werden) zu Angriffszielen führen. Klassiker sind dabei:

    - Bewusst Qualitätsprobleme nicht beseitigen oder einbauen, die später in der Produktion als Angriffsvektor genutzt werden können
    - Einbau von Backdoor-Code, der in der Produktion ausgenutzt werden kann

  - Open Source-Komponenten

    In immer mehr Anwendungen werden Open-Source-Komponenten verbaut, bei denen

nur in den wenigsten Fällen noch eine eigene Qualitätssicherung stattfindet. Man verlässt sich im Allgemeinen auf die „Community“, dass keine gewollten oder ungewollten Sicherheitsbedrohungen enthalten sind. Der kurz vor Jahresende 2021 in großem Umfang ausstrahlende Sicherheitsvorfall der in vielen Anwendungen eingesetzten Java-Komponente „log4j“ zeigt die Gefährdung, die von Open-Source-Komponenten ausgeht, wobei es diesem Fall nicht einmal ein Fehler war, der zum Sicherheitsproblem wurde, sondern ein bewusst flexibel gehaltenes Feature, welches kriminell ausgenutzt werden konnte. Der Schaden allein durch die dadurch bei praktisch allen Unternehmen notwendig werdenden Prüfungen, in welchen eigenen Anwendungen oder auch eingesetzten Standardpaketen die anfällige log4j-Komponente enthalten war, geht in die Milliarden Dollar, nicht zu reden von den Notausschaltungen von tausenden Servern, bei denen man sich nicht sicher war, ob sie gefährdet sind.

Aus allen oben beschriebenen Bedrohungs-Szenarien erwächst eine eigenständige Aufgabe, die heute unter dem Begriff „DevSecOps“ zusammengefasst wird.

Diese muss sich an der derzeit vorherrschenden Sicherheitskultur und -architektur orientieren, die unter dem Begriff „Zero Trust“ bekannt geworden ist.

4 Zero Trust als aktuelle Sicherheitsarchitektur

4.1 Herkömmliche Perimeter-Modelle

In den Zeiten, als Unternehmen noch ihre eigenen Rechenzentren meist mit zentralen Rechnern betrieben und die Anbindung der Benutzer an die Systeme noch weitgehend auf privaten Verbindungen (Standleitungen, Wählleitungen) beruhten, war die „Perimeter-Welt“ noch in Ordnung. Man konnte recht genau sagen, was innerhalb des Perimeters lag und was außerhalb. Das Bild des Rechenzentrums als Burg mit einem Wassergraben drumherum passte noch. Die Netze basierten zumeist noch auf proprietären Standards wie z.B. der „System Network Architecture“ (SNA), in die ein Eindringen von außen sehr schwierig war.

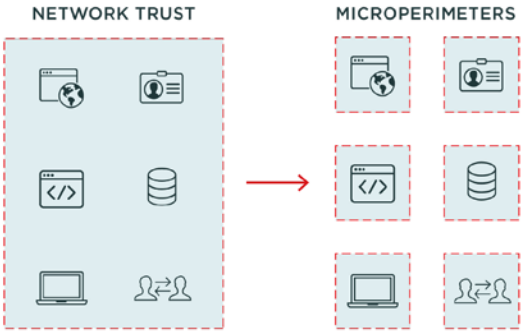


Abbildung 3: Das alte und neue Perimeter-Model (Quelle: pingidentity.com)

Mit der Öffnung der Systeme für Zugänge auf TCP/IP-Basis und dem Aufkommen der Client/Server- und vor allem Cloud-Welten haben sich die Bedrohungen schlagartig erhöht. Man musste das Unternehmensnetzwerk – immer noch im Bild der „Burg“ mit dem Wassergraben“ – in zwei Sphären unterteilen: das innere Netz innerhalb des Perimeters und das äußere Netz

außerhalb des Perimeters. Zusätzlich wurde mit dem zunehmenden E-Mail-Verkehr ein weiteres Einfalltor für Angriffe auf die Unternehmensdaten und -systeme geöffnet.

Auf Basis des Perimeter-Firewalls-Ansatzes werden eine Reihe von üblichen Abwehrmechanismen implementiert:

- Trennung des Netzes in ein internes und ein externes Netz mit einer sog. „Demilitarisierten Zone“ (DMZ) dazwischen
- Schaffung von logischen abgeschirmten Netzen, sog. „Virtual Private Networks“ (VPN), um mit Benutzern außerhalb des Perimeters sicher zu kommunizieren
- Aufbau einer „Intrusion Detection“ auf IP-Adressen-Basis, um Zugriffe aus nicht gewünschten Domains zu unterbinden
- Abwehr gegen „Denial of Service”-Attacken, die versuchen, eine Anwendung durch System-Überlast zu Fehlverhalten zu treiben, das dann für Angriffe ausgenutzt werden kann
- Erkennen von eingehenden E-Mails, die Schadsoftware enthalten (insbesondere in Anhängen)
- Virenscanner, die regelmäßig alle wichtigen Dateien auf Virenbefall prüfen.

Die große Gefahr der obigen Ansätze bleibt, dass ein Angreifer, der sich einmal Zugriff auf den inneren Teil des Netzes erschlichen hat, dort fast nicht mehr aufgehalten werden kann. Dieses kann zwar tlw. durch weitere Mechanismen wie z.B. „Backland“-Firewalls verhindert werden, führt aber in der Regel zum Sicherheitsvorfall.

4.2 Neue Herausforderungen durch neue Systemarchitekturen

In den letzten 10 Jahren hat sich die IT-Welt noch einmal dramatisch verändert, und zwar durch die zunehmende Nutzung von Cloud-Systemen aller Art, sowohl vom Benutzungsmodell als auch vom Verteilungsmodell. So unterscheidet man heute mehr als eine Cloud-Dimension:

- Private Clouds, die noch weitgehend dem klassischen Outsourcing-Modell für Rechenzentren oder auch Anwendungsmanagement durch einen Fremdanbieter entsprechen
- Public Clouds, in denen die Nutzer nur noch Mitnutzer von Systemen sind, die offen am Markt zur Verfügung stehen (die drei großen Anbieter sind Microsoft, Amazon und Google, es gibt aber viele andere Anbieter wie z.B. Salesforce.com, die ebenfalls ihre Systeme in public clouds anbieten).
- Hybride Clouds, bei denen ein Unternehmen – ggf. sogar noch bei Betrieb eines eigenen Rechenzentrums – verschiedene Cloud-Angebote parallel nutzt.

In der weiteren Dimension geht es um den Umfang der Services, die man von einem Cloud-Anbieter (sei es private oder public) nutzt:

- Entspricht das IaaS (Infrastructure as a Service)-Modell noch einem outgesourcten Rechenzentrum, das man weitgehend noch selbst managt und betreibt, so
- hat man beim PaaS (Platform as a Service)-Modell schon Teile des Basisbetriebs (von der Hardware über Betriebssystem bis zur Middleware) in die Verantwortung des Cloud-Betreibers gegeben.
- Beim SaaS (Software as a Service)-Modell letztlich nutzt man fertige Anwendungen des Cloud-Anbieters, die man in gewissem Umfang für eigene Anforderungen customizen kann, bei denen aber der Cloud-Anbieter den gesamten Entwicklungs- und Tagesbetrieb in den Händen hat.

Auf die rechtlichen Anforderungen aus dieser Verantwortungsabgabe gerade im Hinblick auf die EU-DSGVO soll hier nicht eingegangen werden. Es wird auf die einschlägige Literatur insbesondere vom BSI verwiesen.

In der Regel nutzen große Unternehmen heute ein Gemisch aus den oben dargestellten Modellen. Dadurch, dass auch Fachbereiche selbstständig (und manchmal leider auch unabgestimmt) Cloud-Services ordern können, wird das Management der Systemlandschaft tendenziell immer komplizierter und damit angreifbarer.

Es versteht sich fast von selbst, dass in einem solchen Systemumfeld der klassische Perimeter-Begriff nicht mehr zeitgemäß ist. Er wird heute daher durch viele „Mini-Perimeter" ersetzt, die eine neue Sicherheitsarchitektur und auch -kultur erfordern. Eine wesentliche ist die sog. „Zero Trust"-Architektur, die im nächsten Kapitel vorgestellt wird.

4.3 Definition und Grundprinzipien von Zero Trust

Der Begriff „Zero Trust" im Zusammenhang mit IT-Sicherheitsarchitekturen ist nicht neu. Nach ersten Anfängen in den 2000er Jahren durch das amerikanische Militär, von einer perimeterbezogenen zu einer eher transaktionsbezogenen Absicherung zu wechseln (unter dem Begriff „black core" veröffentlicht), ist dieser Ansatz dann u.a. von Forrester Research aufgegriffen worden. Der Begriff „Zero Trust" geht zurück auf eine Serie von Reports, die John Kindervag zusammen mit weiteren Forrester-Kollegen seit 2010 [FORREST 2010] erarbeitet hat und die dann im April 2013 als Antwort auf einen RfI des NIST [NIST 2013] erstmalig zusammengefasst dargestellt wurden.

Mittlerweile existiert ein eigenes NIST-Standard-Dokument [NIST 2020], das Zero Trust-Architekturen detailliert beschreibt, und damit die derzeit wesentliche Referenz zu diesem Thema darstellt.

[NIST 2020] definiert „Zero Trust"-Architekturen wie folgt (direkte Übersetzung):

„Zero Trust (ZT) bietet eine Sammlung von Konzepten und Ideen, die darauf abzielen, die Unsicherheit bei der Durchsetzung präziser Entscheidungen über den Zugang zu Informationssystemen und -diensten mit den geringsten Privilegien pro Anfrage angesichts eines als kompromittiert betrachteten Netzwerks zu minimieren. Eine Zero Trust Architecture (ZTA) ist der Cyber-Sicherheitsplan eines Unternehmens, der Zero Trust-Konzepte verwendet und Komponentenbeziehungen, Workflow-Planung und Zugriffsrichtlinien umfasst. Ein „Null-Vertrauen"-Unternehmen besitzt daher die Netzwerkinfrastruktur (physisch und virtuell) und die betrieblichen Richtlinien, die für ein Unternehmen als Produkt eines Null-Vertrauen-Architekturplans gelten."

Konkret umgesetzt heißt dieses nach [NIST 2020], dass folgende Grundsätze in einer Zero Trust-Architektur gelten:

- Alle Datenquellen und sonstigen IT-Einrichtungen werden einheitlich als „Ressource" gemanagt. Dabei können auch Endgeräte, die z.B. SaaS-Services nutzen, als Ressource definiert werden, so wie der Service selbst auch.
- Jede Form der Kommunikation wird abgesichert, unabhängig davon, woher aus dem Netzwerk sie initiiert wird. Die Zugehörigkeit eines Geräts zu einem bestimmten Netzwerksegment (also z.B. innerhalb des Perimeters im klassischen Sinn) reicht nicht mehr allein aus, um einen Ressource-Zugriff zu gewähren.
- Der Zugriff auf alle Unternehmensressourcen (im obigen Sinn) wird nur auf Session-Basis gewährt. Jeder Anforderer einer Ressource wird bei jeder Anforderung aktuell geprüft, ob er in Bezug auf den Ressourcenzugriff als vertrauenswürdig gilt.
- Der Zugang zu Ressourcen wird durch dynamische Richtlinien bestimmt - einschließlich des beobachtbaren Zustands der Benutzeridentität, der Anwendung/des Dienstes und des anfordernden Devices - und kann weitere Verhaltens- und Umweltattribute umfassen.
- Das Unternehmen überwacht und misst die Integrität und die Sicherheitslage aller eigenen und damit verbundenen Assets. Keinem Asset – egal welcher Art – wird automatisch vertraut.

- Jede Ressourcen-Authentifizierung und -Autorisierung ist dynamisch und wird rigide durchgesetzt, bevor ein Zugriff gestattet wird. Damit entsteht ein ständiger Kreislauf aus Zugriffserteilung, Scannen und Bewertung von Bedrohungen, Anpassung und ständiger Neubewertung des Vertrauens in die laufende Kommunikation.
- Das Unternehmen sammelt so viele Informationen wie möglich über den aktuellen Zustand aller Ressourcen, der Netzwerkinfrastruktur und der Kommunikation und nutzt sie zur stetigen Verbesserung der Sicherheitslage. Diese Daten müssen entsprechend ausgewertet werden und dienen zur Verbesserung der Erstellung und Durchsetzung der Richtlinien.

Aus den oben genannten Grundsätzen leitet sich ab, dass in einer Zero-Trust-Architektur folgende Objekte eine zentrale Rolle für die dynamische Bewertung von Zugriffen auf jegliche Form von Ressource – und dabei stehen die eigentlichen Daten natürlicherweise im Zentrum – spielen:

- Benutzer

  Die Benutzer eines Systems stehen auch bei Zero Trust-Architekturen im Zentrum der IT-Sicherheit. Ihre Authentifizierung ist dabei ebenso kritisch wie ihre Autorisierung für Zugriff auf bestimmte Ressourcen. Der Begriff „Role Based Access Control“ (RBAC) spielt dabei eine zentrale Rolle und wird u.a. auch von der Regulatorik eingefordert: Jeder Benutzer sollte nur genau die Zugriffsrechte auf Ressourcen haben, die er aufgrund seiner Funktion im Unternehmen benötigt und keine weiteren. Vertritt er eine andere Person, so ist dem Benutzer temporär deren Rolle zuzuweisen. Diese Verwaltung ist naturgemäß aufwendig und wird heute durch spezielle Systeme unter dem Namen „Identity and Access Management“ (IAM) unterstützt. Diese unterstützen nicht nur „Single Signon“-Ansätze (bei denen der Nutzer die spezifischen Zugriffs-Informationen für die einzelnen Systeme gar nicht mehr kennen muss), sondern auch die Mehrfach-Authentifizierung (MFA) beim Anmelden am zentralen Authentifizierungs-Server.

  Eine besondere Rolle spielen die privilegierten Benutzer eines IT-Systems, also Administrationen und andere „Super-Benutzer“ mit erweiterten Zugriffsrechten. Neben der Absicherung schon bei der Einstellung (im Sinne des Geschäftsgeheimnis-Gesetzes, s.o.) werden für diese Nutzer mittlerweile auch über IAM-Funktionen hinausgehende Sicherungssysteme unter dem Begriff „Privileged Account Management“ (PAM) eingesetzt, die z.B. Vier-Augen-Prinzipien o.ä. bei Benutzer- und Systemanpassungen sicherstellen.

  Aus Sicht von Zero Trust-Architekturen sind insbesondere sog. „Technische User Accounts“, die keiner Person zugeordnet sind, sondern z.B. in API-Aufrufen von Dritt-Systemen genutzt werden, die keine eigene Prüfung mehr durchführen, eigentlich nicht mehr tragbar, da sie ein „offenes Tor“ zu diesem System bieten, falls jemand an die Anmeldeinformationen dieser „Benutzer“ gelangen sollte.

- Gerät und Plattform

  Neben der Information über den Anforderer einer Ressource ist es in Zero Trust-Architekturen ebenso wichtig zu wissen, von welchem Device (Rechner, Smartphone, IoT-Gerät, etc.) eine Anforderung geschickt wird.

  Dazu gehören nicht nur die Geräte-Kennung (klassisch immer noch MAC-Adressen), sondern auch die Information über den genutzten Browser und weitere Plattforminformationen.

- Lokation und Zeit

  Gerade in der Welt des Internet-Zugriffs ist es für die Bewertung eines Zugriffs in Bezug auf seine Zulässigkeit unerlässlich, die aktuelle Lokation des Anforderers zu kennen.

  Diese kann zusammen mit der Tageszeit – sowohl am aktuellen Ort (auf Basis der IP-Adresse) des Requests als auch am Standort der Ressource selbst – Aufschluss darüber geben,

ob ein Request sachlich logisch sein kann oder ob es sich um einen Eindringversuch eines nicht berechtigten Dritten handeln könnte.

- Spezifisches Risiko des Zugriffs

  Abhängig von der Risikoeinstufung der angeforderten Ressource können weitegehende Bewertungen erforderlich werden, die dann entsprechende Aktionen auslösen können von der Warnmeldung an den Anforderer „Wir sehen, was Du hier machen willst. Brauchst Du diesen Zugriff wirklich?“ über stille Alarme, die in einer Sicherheitszentrale („Security Operation Center“ – SOC) auflaufen bis hin zur sofortigen Sperrung des Zugriffs und ggf. sogar des gesamten Benutzer-Accounts. Diese Form der Risikobewertung erfordert eine vorherige Einstufung aller kritischen Assets in entsprechende Risikoklassen.

5 DevSecOps als konkrete Umsetzung von Zero Trust

5.1 Cybersecurity-Architekturen als Grundlage der Abwehrorganisation

Bei der Umsetzung von Zero Trust gibt es nicht die „eine“ Lösung, sondern jedes Unternehmen muss eine entsprechende Analyse der potentiellen Bedrohungslage und vor allem der wesentlich zu schützenden Ressourcen (letztlich die Daten und Datenobjekte unterschiedlicher Art) durchführen.

Die Umsetzung auch von Zero Trust-Ansätzen folgt i.d.R. bekannten Frameworks von NIST oder dem BSI, sehr anschaulich ist auch die Security Architecture von Gartner [GARTNER 2018] für die Einordnung der einzelnen Aktivitäten:

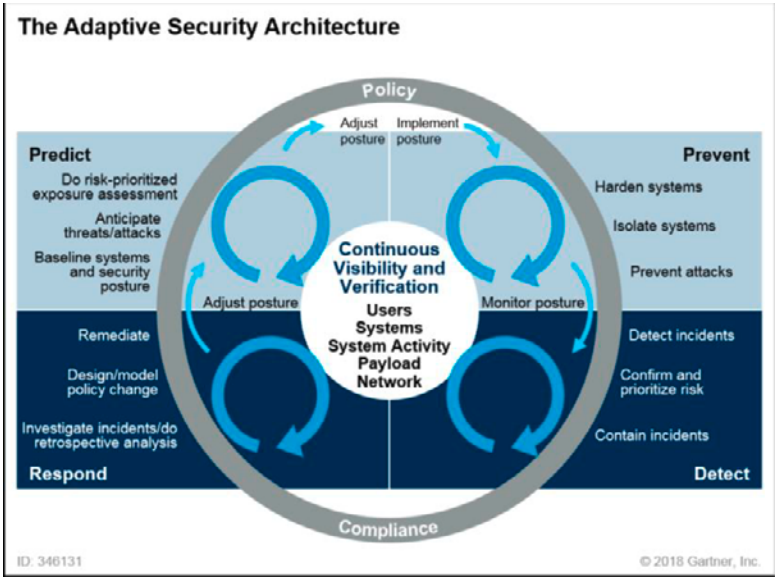


Abbildung 4: Aufbau einer Security-Architektur nach [GARTNER 2018]

Predict: Die Aktivitäten zum Aufbau einer Zero Trust-Architektur beginnen dabei mit einer Analyse der Bedrohungslage und Klassifikation der zu schützenden Assets.

Prevent: Hierzu gehören die klassischen Abwehrmaßnahmen, die man auch aus perimeterorientierten Sicherheitsarchitekturen kennt und die natürlich nicht völlig obsolet sind, sondern die äußere Schale der Sicherheits-„Zwiebel“ bilden. Dazu gehören die klassischen Ansätze wie:

- Router
- Firewalls und demilitarisierte Zonen
- Intrusion Detection
- Intrusion Prevention
- Endpoint Protection Systeme
- Malware-Erkennung und -abwehr inkl. Phishing-Erkennung

Auch die gesamten Systeme zur Verwaltung der Benutzer-Authentifizierung und Autorisierung gehören in diesen Quadranten.

Detect: Dieser Bereich enthält die Systeme, die besonders den Zero Trust-Ansatz unterstützen. Sie bewerten das Benutzerverhalten und versuchen, auf Basis der bekannten Verhaltensweisen der Benutzer, abweichende und damit verdächtige Aktivitäten zu identifizieren und ggf. auch direkt zu verhindern. Hierzu werden heute bereits auch Ansätze aus dem „Machine Learning“ eingesetzt, um eine verfeinerte Analyse durchzuführen und damit insbesondere zu viele „false positives“-Alarme in den entsprechenden Überwachungssystemen (SIEM) zu verhindern. In diese Kategorie gehören Systeme wie

- SIEM – „Security Information and Event Management“ : Überwachung aller Logfiles und Monitoring-Ausgaben gerade im Bereich Login, Zugriffe etc. Automatische Auslösung von sog. Security Incidents bei Auffälligkeiten.
- UEBA – „User and Entity Behavioral Analytics”: Unter diesem Begriff werden alle Prozesse und Systeme zusammengefasst, die das Unternehmen vor Angriffen von innen schützen. Dazu werden die bekannten und zugelassenen Verhaltensweisen in Bezug auf die Benutzer („User“) und Ressourcen („Entity“) zur Bewertung jedes Zugriffsversuchs herangezogen, zunehmend auch mit Methoden des Machine Learnings.
- DLP – Data Loss (auch: Leakage) Prevention: Eine konkrete Umsetzung innerhalb von UEBA. Dabei erfolgt eine Bewertung jedes Datenobjekt-Zugriffs auf seine Plausibilität und Notwendigkeit mit entsprechend abgestuften Reaktionsmöglichkeiten.
- TIS – Threat Intelligence Service: Ein Threat Intelligence Service liefert aktuelle Informationen zur bekannten Bedrohungslage der IT-Sicherheit. Hierfür sammelt der Service Daten aus unterschiedlichen Quellen (bis hinein in das Dark Web) und stellt sie in aufbereiteter Form zu Verfügung. Insbesondere die Suche nach gestohlenen Zugangsdaten stellt einen wesentlichen Teil der Services dar.

Respond: Dieser Bereich enthält alle Prozesse und Systeme, die im Fall eines erfolgten Angriffs aktiviert werden müssen. Dabei steht zunächst die Schadensbegrenzung im Vordergrund, danach folgt die Wiederherstellung der ggf. korrumpierten Daten (durch hoffentlich aktuelle und auch noch zugreifbare Backups) und letztlich die Forensik, um zu erkunden, wie der Angriff gelingen konnte.

5.2 DevSecOps als Umsetzungsmodell im Entwicklungsbereich

Unter dem Begriff DevSecOps werden alle Maßnahmen zusammengefasst, die im Entwicklungs- und vor allem in automatisierten Deployment-Prozessen eine Rolle spielen und dabei die Grundpfeiler der Zero-Trust-Architektur in konkrete Handlungs- und Automatisierungsansätze umsetzen.

Dazu gehören die in Kap. 5.1 schon genannten Bereiche, die auch für die Produktion gelten:

- Starke Authentifizierungs- und Autorisierungsmechanismen,
  - heute i.d.R mit Multi-Faktor-Authentifizierung

- Network Access Control bzw. Software Defined Perimeters
  - d.h. maximale Abschottung von Produktion und Nicht-Produktion
  - Zonen und Cluster (z. B. durch Definition von Sicherheitsgruppen)
- Logging, Monitoring, Auditing
  - heute zumeist abgebildet in SIEM-Systemen, die bei Verletzungen von Compliance-Regeln sofort alarmieren
- Verhinderung von unerwünschten Informationsabflüssen über Nicht-Produktions-Umgebungen, heute in Data Leak-Prevention-Tools

Hinzu kommen die für Entwicklung und Deployment wesentlichen Absicherungen:

- Besondere Absicherungsmaßnahmen in Bezug auf die privilegierten Benutzer speziell im Deployment-Bereich
- Peer Reviews und statische Code-Analyse der entwickelten Software
  - Entsprechende Tools stehen zur Verfügung
- Absicherung der eingesetzten Open-Source-Komponenten
  - Auch hierzu existiert eine Tool-Unterstützung, bekannt ist vor allem das Produkt „Black Duck“, welches auch Compliance-Prüfungen bzgl. lizenzrechtlicher Aspekte mit durchführt
- Das ohnehin in vielen Bereichen von der Regulatorik geforderte „Vier-Augen-Prinzip“ auch in der Deployment-Tool-Kette vorsehen und entsprechende „Interrupts“ einbauen, bei denen zumindest ein zweiter Benutzer die weiteren Prozessschritte freigeben muss.

Literatur

[ALLSHAM 2009] John Allspaw and Paul Hammond: 10+ Deploys per Day - Dev and Ops Cooperation at Flickr, Velocity 09 – O'Reilly Conferences, 22.-24.06.2009, Download am 23.02.2018, https://conferences.oreilly.com/velocity/velocity2009/public/schedule/detail/7641

[BITKOM 2018] Bitkom: Spionage, Sabotage und Datendiebstahl – Wirtschaftsschutz in der Industrie, Bitkom e.V., Studienbericht 2018, https://www.bitkom.org/sites/default/files/file/import/181008-Bitkom-Studie-Wirtschaftsschutz-2018-NEU.pdf (letzter Zugriff: 06.10.2020)

[BKA 2017] BKA: Innentäter in Unternehmen, November 2017, https://www.wirtschaftsschutz.info/SharedDocs/Artikel/DE/BKA-Monitoringbericht-Innentaeter.html;jsessionid=E68CECFB30743415006344B2D1592D44.2_cid349?nn=6555960 (letzter Zugriff: 06.10.2020)

[BORCH 2018] Borchers, J.: Agile und DevOps treffen auf Regulatorik – Konfrontation oder Kooperation? , GI Rundbrief des Fachausschusses Management der Anwendungsentwicklung, Jahrgang 24, Heft 1, April 2018, ISSN 1610-5753

[BORCH 2019] Borchers, J.: Compliance und IT-Security - Kommt die Bedrohung immer von außen?, ECC2019, https://cecmg.de/download/events/2019/ECC/inhalt/praesentationen/ECC2019_Vortrag_03_IT-Security_Keynote_Jens_Borchers.pdf , (letzter Zugriff: 05.10.2020)

[BORCH 2020] Borchers, J.: Zero Trust-Architektur und -Kultur,
Andreas Schmietendorf, Konrad Nadobny (Hrsg): ESAPI 2020, 4. Workshop Evaluation of Service-API, Shaker Verlag 2020, ISBN 978-3-8440-7515-1

[BSI 2021] BSI: Die Lage der IT-Sicherheit in Deutschland 2021,
https://www.bmi.bund.de/SharedDocs/downloads/DE/publikationen/themen/it-digitalpolitik/bsi-lagebericht-cybersicherheit-2021.pdf (letzter Zugriff: 01.01.2022)

[FORREST 2010] Kindervag, J. et.al: No More Chewy Centers - The Zero Trust Model Of Information Security, Forrester Report, 2010, aktuelle Version: https://www.forrester.com/report/No+More+Chewy+Centers+The+Zero+Trust+Model+Of+Information+Security/-/E-RES56682# (letzter Zugriff: 06.10.2020)

[GARTNER 2018] Gartner: Market Guide for Endpoint Detection and Response Solutions, November 2018

[NIST 2013] NIST: Developing a Framework to Improve Critical Infrastructure Cybersecurity, RFI# 130208119-3119-01, 08.04.2013, https://www.nist.gov/system/files/documents/2017/06/05/040813_forrester_research.pdf, (letzter Zugriff: 06.10.2020)

[NIST 2020] NIST: Zero Trust Architecture, NIST Special Publication 800-207, August 2020, https://doi.org/10.6028/NIST.SP.800-207 (letzter Zugriff: 06.10.2020)

[PAUL 2014] Fredric Paul: The Incredible True Story of How DevOps Got Its Name, New Relic Blog, Download am 23.02.2018, https://blog.newrelic.com/2014/05/16/devops-name/

[SIMON 2020] Simon, F.: Digitalisierung: Freund und Feind der Security“, Keynote auf der Enterprise Computing Conference ECC 2020, Köln, 13.03.2020, https://cecmg.de/download/events/2020/ECC/ECC2020_Ticker_Security%20Workshop_Agenda_V2.pdf

Usability from a Product Manager's Perspective[1]

Hans-Bernd Kittlaus

Managing Director of InnoTivum, Chairman of ISPMA® e.V.

In the last 20 years, usability has turned into a key success factor for software-intensive products. Successful product companies like Apple have demonstrated the power of excellent user experience design (UX design). Today any software product manager needs to focus on the usability of her/his product, certainly for consumer products, but more and more also for enterprise products.

A software product manager is responsible for managing software with the objective to achieve sustainable success over the life cycle of a software product or software parts of software-intensive products, i.e. systems or services. This generally refers to economic success, which is ultimately reflected by the profits generated. Software product managers have the business responsibility across different versions, variants and associated services of a product. They have to manage a broad set of product-related activities as shown in the ISPMA® SPM Framework (Figure 1). They have to act proactively and be the responsible and engaged driver of their products.

It has become quite obvious that good usability has a direct impact on product success. Purdue University's Kyungdoh Kim e.a. proved this for cell phones in 2012 already ([Kim, Proctor & Salvendy 2012]).

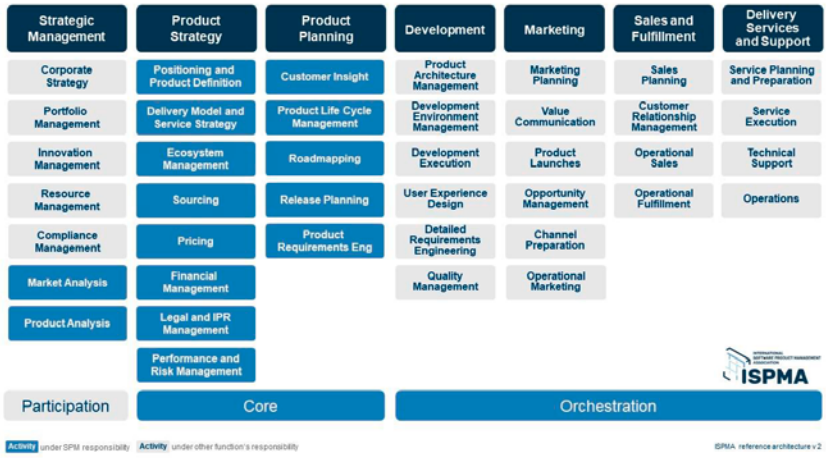


Figure 1: ISPMA® SPM Framework V.2.0

The SPM framework provides a holistic view on the activities of software product management. It is structured horizontally (columns) based on the functional areas of a software organization. There is an additional overlay structure with "Core SPM", "Participation" and "Orchestration".

[1] Fachbeitrag erschien ursprünglich in: SQ Mag (The Voice of Software Quality) Nr. 10 Oktober 2021 (S. 4 und 5), iSQI https://www.sq-mag.com/about/

Activities under Orchestration are under the responsibility of the respective functions. However, the activity of Orchestration itself is a core responsibility of SPM.

UX Design is functionally placed in the Development column which does not necessarily mean that it needs to be part of a software development unit. Often UX Design is organized as a shared-resource department that works on products that belong to different product units. Some companies focus more on the requirements perspective and put UX Design close to Product Management, some focus on the implementation aspect and keep it closer to Development. In any case, tight cooperation of UX Design and Product Management must be ensured.

"User Experience (UX) design can be a key factor for differentiation and competitive strength. It addresses every aspect of the users' interactions with a software product or component with the purpose of shaping the user's behaviors, attitudes, and emotions about that product or component." It is much broader than just usability, as it is "... covering or interacting with disciplines like graphic design, information architecture, Human-Computer-Interface (HCI) design, interaction design and usability engineering." *(ISPMA® FL (2021))*

M. Cagan (2013) distinguishes four design-related activities that are critical to the success of software products: interaction design, visual design, rapid prototyping and usability testing. These four roles need to "... work closely with the (software) product manager to discover the blend of requirements and design that meet the needs of the user." *(M. Cagan (2013))*.

"Due to the objectives of UX Design, there is a significant overlap with the product manager role, especially in the following areas:

- Developing a deep understanding of customers' real needs
- Understanding intended product usage
- Developing product scope and product definition
- Eliciting high-level product requirements

In these areas, software product managers may find that UX designers are powerful allies that help them define a product that serves customers and users even better - or they might be in stark conflict, quarreling over decisions and accountabilities." *(ISPMA® EL (2021))*

Recently, there have been attempts to combine the roles of product manager and UX Designer into one. While this may be conceptually attractive, we consider it as wishful thinking since it is difficult enough to be an excellent UX designer or an excellent product manager. How unlikely is it to find that super(wo)man who manages to be excellent in both?

"A software product manager is well advised to canalize the creativity of UX designers into the refinement of early product concepts and utilize their experimentation skills to get evidence that the product concept works for the intended users. If the UX designers discover significant problems in user acceptance and product effectiveness, the product manager may have to pivot the product concept." *(H.-B. Kittlaus & S. Fricker (2017))*.

The better the cooperation between Product Management and UX Design works, the higher the chances for product success.

Literature

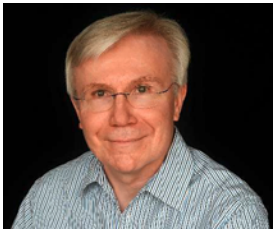
Marty Cagan (2018): INSPIRED: How to Create Tech Products Customers Love, 2nd edition, Wiley

ISPMA® EL (2021): ISPMA® SPM Excellence Level Orchestration syllabus V.2.0 (www.ispma.org)

ISPMA® FL (2021): ISPMA® SPM Foundation Level syllabus V.2.0 (www.ispma.org)

Kyungdoh Kim, Robert W. Proctor & Gavriel Salvendy (2012): The relation between usability and product success in cell phones, Behaviour & Information Technology, 31:10, 969-982

H.-B. Kittlaus & S. Fricker (2017): Software Product Management, The ISPMA®-Compliant Study Guide and Handbook, Springer



Hans-Bernd Kittlaus is an internationally renowned expert on SPM and a highly experienced SPM trainer and consultant. He has been working for software organizations of all sizes, and runs his own company **InnoTivum** (www.innotivum.com). Before he was head of SPM and development units of IBM. He is the chairman of ISPMA®, and has published numerous articles and books, the latest being "Software Product Management" (see Literature).

INTERNATIONAL
SOFTWARE PRODUCT MANAGEMENT
ASSOCIATION
ISPMA

The International Software Product Management Association (**ISPMA e.V.**, www.ispma.org) is a group of SPM experts from academia and industry that aims at fostering software product management excellence across industries by establishing software product management as a discipline of its own in both academia and industry. ISPMA provides a curriculum with Foundation and Excellence Level training modules and corresponding certification which are frequently updated.

Handlungsempfehlungen für die Erstellung einer Zielarchitektur zur Integration von IT-Anwendungen

Georgios Kostakopoulos, Claudia Hess, Marian Benner-Wickner

IU Internationale Hochschule

marian.benner-wickner@iu.org

Abstract: Das kontinuierliche Wachstum von IT-Anwendungslandschaften ist durch die Vielzahl an technologischen Neuentwicklungen nicht zu bremsen. In der IT großer Unternehmen gehört es daher zum Tagesgeschäft, neue IT-Anwendungen mit EAI-Technologien zu integrieren und bestehende IT-Anwendungslandschaften mit Konzepten, Methoden und Werkzeugen des EAM am Leben zu erhalten. Umso erstaunlicher ist es, dass beide Themen – EAM und EAI – in der Fachdiskussion nur selten gemeinsam betrachtet werden. Dieser Beitrag schließt diese Lücke, indem praxiserprobte Handlungsempfehlungen für die Erstellung einer Zielarchitektur zur Integration von IT-Anwendungen vorgestellt werden. Sie setzen auf die etablierten EAM-Zielmuster von Keller auf und konkretisieren diese hinsichtlich der IT-Anwendungsintegration. Anhand einer Fallstudie und eines Applicability Checks konnte gezeigt werden, wie die Handlungsempfehlungen Unternehmen dabei unterstützen, eine nützliche, konsistente und einfache Zielarchitektur zu erarbeiten.

1 Einleitung

Im digitalen Zeitalter steigt die Wichtigkeit der *Integration von IT-Anwendungen* (engl. *Enterprise Application Integration, EAI)*[1] kontinuierlich. Sie gilt als Voraussetzung für die Erreichung der Unternehmensziele, u.a. weil dadurch Themen wie Automatisierung, Flexibilität und Orientierung an den Wünschen der Kundschaft unterstützt werden [HRZ21], [LU13]. Bekanntermaßen sind beispielsweise Kundschaftsdaten eines Unternehmens, welche typischerweise zentral im Enterprise Resource Planning (ERP)-System gepflegt werden, ebenfalls ein wichtiger Bestandteil des Customer Relationship Management (CRM)-Systems des Unternehmens, um dort wiederum die Kundschaftsbesuche zu planen und den jeweiligen Umsatz analysieren zu können. Ebenso ist es in vielen Unternehmen erforderlich, diese Daten dem Außendienst-Personal mobil zur Verfügung zu stellen, um bei einem Besuch vor Ort darauf zugreifen zu können. Weiterhin müssen diese Daten bzw. ein Teilbereich davon nicht selten externen Dienstleistern übermittelt werden, zum Beispiel um das Kaufverhalten zu analysieren.

Solche Beispiele sind stark vereinfacht, denn heutzutage haben große Unternehmen eine dreistellige Zahl an IT-Anwendungen im Einsatz, in den meisten Fällen verursacht durch historisch bedingtes Unternehmenswachstum, den damit verbundenen Anforderungen und die Nichtablösung von Altsystemen infolge der vorhandenen Abhängigkeiten und Risiken [Be14], [Ti11]. Wong spricht dabei sogar von mehreren tausend IT-Anwendungen bei globalen Unternehmen [Wo18]. Diese befinden sich in unterschiedlichen IT-Lebenszyklusphasen, verwenden unterschiedliche Technologien und wurden von unterschiedlichen Softwarehäusern entwickelt. Neben den internen IT-Anwendungen, die miteinander integriert werden müssen, wird heutzutage der Bedarf nach Integration mit externen IT-Anwendungen immer größer. Damit sind nicht nur IT-Anwendungen der Kundschaft, der liefernden Unternehmen oder Behörden gemeint, sondern auch eigene IT-Anwendungen, die in der Cloud gehostet werden. Es wird erwartet, dass weitere Entwicklungen in den Bereichen Internet of Things (IoT), Cloud und Mobile Com-

[1] Der Begriff Integration bzw. IT-Integration bezieht sich auf die Zusammenführung von IT-Anwendungen, die unabhängig voneinander entwickelt wurden. Dadurch soll die Gewinnung neuer Funktionalitäten sowie eine Effizienzsteigerung ermöglicht werden [IB20], [Me13].

puting neue Herausforderungen für die Integration von IT-Anwendungen mit sich bringen werden [HRZ21], [RMR17]. Eine EAI-Lösung ist dabei als „modularer Werkzeugkasten“ zu verstehen [AS05]. Dabei wurden über den Lauf der Jahre verschiedene technische Ansätze entwickelt, wie verschiedene IT-Anwendungen zu einer ganzheitlichen Lösung kombiniert werden können, z.B. durch den Einsatz von Integrationssystemen oder von APIs.

Dass das Thema IT-Anwendungsintegration für ein Unternehmen sehr komplex und herausfordernd sein kann, ist keine neue Erkenntnis [HW03], [LU13], [SA12]. Für viele Unternehmen ist jedoch die Erarbeitung eines Zielbilds für EAI schwierig, da dies durch eine hohe Anzahl an unterschiedlichen Integrationstechnologien und -ansätzen maßgeblich erschwert wird [Mü05]. Dies gilt auch heute noch und wird durch die schnellen technologischen Änderungen sowie den sich schnell verändernden Märkten und Rahmenbedingungen weiter verstärkt (z.B. [Ju21]). Hinzu kommen die vielfältigen Bedarfe der verschiedenen Stakeholder im Unternehmen, die unter Umständen auch in Widerspruch zueinanderstehen, z.B. hinsichtlich Qualitäts- und Compliance-Anforderungen, Prioritäten und Budgets [ADS21]. Diese Herausforderungen erschweren eine unternehmensweite Betrachtung der IT-Anwendungslandschaft. Fehlt ein übergreifendes Konzept zum systematischen Management der IT-Landschaft in Bezug auf die IT-Anwendungsintegration, so kann der steigenden Komplexität nicht gezielt entgegengewirkt werden [SA12]. Dies führt zu Redundanzen, Inkonsistenzen und einer fehlenden Übersicht über die eingesetzten Integrationsansätze bzw. Schnittstellen. In einer solchen Situation ist es kaum noch möglich, Synergieeffekte auszunutzen (z.B. [Be14], [Ti11]). Mit seinen Methoden und Werkzeugen bietet das Enterprise Architecture Management (EAM) einen Ansatz, um genau diese Herausforderungen bei der IT-Anwendungsintegration zu adressieren [ADS21]. Für die Erarbeitung eines Zielbilds für die IT-Anwendungsintegration spielen insbesondere die Prozesse und Methoden der strategischen Bebauungsplanung eine zentrale Rolle. In diesem Rahmen wird der zukünftige, erstrebenswerte Zustand der IT-Anwendungslandschaft erarbeitet, die sogenannte Zielarchitektur [Th18].

Im Mittelpunkt dieses Beitrags steht deshalb die methodische Unterstützung der Erstellung einer Zielarchitektur speziell im Hinblick auf die Integration von IT-Anwendungen. Es werden Handlungsempfehlungen erarbeitet, die es Unternehmen ermöglichen, ein Zielbild für die IT-Anwendungsintegration zu entwickeln. Die vorgestellten Handlungsempfehlungen sollen sowohl präzise sein als auch die unternehmensspezifische Ausprägung ermöglichen. Mit diesen Handlungsempfehlungen zielt dieser Artikel darauf ab, dass EAM-Teams auf Erfahrungen und bewährte Lösungsmuster zugreifen können.

Zur Erarbeitung der Handlungsempfehlungen für die IT-Anwendungsintegration wird auf die Zielmuster von Keller Bezug genommen. Diese stellen „typische Anforderungen an die IT-Funktionen eines Unternehmens“ [Ke17], S. 7 dar, wobei in der Praxis in der Regel mehrere Zielmuster gleichzeitig relevant sind. Generell erleichtert die Orientierung an solchen Zielmustern die unternehmensspezifische Ausprägung des EAM, denn abhängig von den Zielsetzungen und Prioritäten des Unternehmens können die relevanten Zielmuster ausgewählt und unternehmensspezifisch ausgeprägt werden [Ke17].

2 Verwandte Arbeiten

Um den aktuellen Stand der relevanten Lösungsansätze aus den Themenbereichen EAM und EAI zu ermitteln und vor allem um die explizite Verbindung beider Themen in der wissenschaftlichen Diskussion zu untersuchen, wurde eine systematische Literaturanalyse nach Webster und Watson durchgeführt [WW02]. Dabei wurden wissenschaftliche Publikationen ausfindig gemacht, welche zwischen 2016 und 2021 veröffentlicht wurden und die IT-Anwendungsintegration aus der EAM-Perspektive betrachten. Darüber hinaus wurde die Literatursuche, wo immer es möglich war, durch eine Rückwärts- und Vorwärtssuche ergänzt [WW02].

Die Literaturanalyse zeigte, dass die Verbindung der zwei Themenbereiche EAM und EAI bislang nur unzureichend integrativ betrachtet ist. Die identifizierten relevanten Publikationen sind zum einen EAM-Ansätze, bei denen EAI nur sehr oberflächlich betrachtet wird: Keller liefert eine umfassende Betrachtung von EAM mithilfe eines musterbasierten Ansatzes [Ke17], während Hanschke eine agile Methode für eine schrittweise Einführung von EAM-Konzepten in Unternehmen entwickelt [Ha13]. Im Rahmen zweier weiterer Publikationen gilt EAM als Mittel, um eine Digitalisierungsstrategie in einem Krankenhaus zu implementieren und somit die internen Abläufe zu verbessern [MB18], [MB20]. Diese Publikationen erwähnen EAI als Ansatz, der für den Datenaustauch zwischen IT-Anwendungen Hilfestellung bieten kann, liefern jedoch keine Details. Zum anderen gibt es Publikationen, welche EAI aus einer technischen Perspektive betrachten: Hohpe und Woolf fokussieren sich auf sogenannte Enterprise Integration Patterns, die verwendet werden, um im Rahmen von Unternehmensarchitekturen die lose Kopplung von IT-Anwendungen zu erzielen [HW03]. Dieser Ansatz wird einige Jahre später von Ritter et al. weiterentwickelt bzw. um weitere Patterns ergänzt [RMR17]. Des Weiteren wird EAI in einer empirischen Studie von Aier und Schönherr als integrierendes Element von Unternehmensarchitekturen angesehen, welches sowohl IT-bezogene Elemente untereinander als auch fachliche und IT-bezogene Elemente miteinander verbindet. Die Autoren formulierten somit schon vor einigen Jahren Handlungsempfehlungen für ein erfolgreiches Management von Integrationsarchitekturen, sie weisen aber auf den explorativen Charakter ihrer Studie hin und betonen den Bedarf nach mehr Forschung [AS05]. Dazu muss ergänzt werden, dass sich seit dem Erscheinen der Studie die für die IT-Anwendungsintegration eingesetzten Technologien stark verändert haben und die erarbeiteten Handlungsempfehlungen demnach aktuelle Herausforderungen nicht berücksichtigen.

Im Kontext des EAM ist es zudem wichtig, zu prüfen, welche Methoden und Ansätze die sogenannten EAM-Frameworks für die IT-Anwendungsintegration bieten. Frameworks wie The Open Group Architecture Framework (TOGAF) dienen vielen Architekturteams als Blaupause und Nachschlagewerk. TOGAF adressiert die IT-Anwendungsintegration nur in einem größeren Kontext und bietet Leitlinien, wie Anforderungen an die Interoperabilität in den verschiedenen Phasen einer Architekturinitiative berücksichtigt werden können [Th18]. Diese Interoperabilitätsanforderungen sollen gemäß TOGAF – wie auch in diesem Beitrag dargestellt – in der Zielarchitektur verankert werden. Eine detaillierte Ausprägung dieser Anforderungen bzw. konkrete Handlungsempfehlungen für die IT-Anwendungsintegration sind nicht Bestandteil von TOGAF.

3 Handlungsempfehlungen

Für die Erarbeitung der Handlungsempfehlungen zur Erstellung einer Zielarchitektur für die IT-Anwendungsintegration dienen ausgewählte Zielmuster nach Keller als Grundlage. Als Teil der strategischen Bebauungsplanung sind folgende Zielmuster besonders relevant [Ke17]:

- Optimierung mit Sourcing-Strategien,
- Verbesserung Time-to-Market,
- Reduktion von Heterogenität, und
- Bewältigung von Fusionen.

Für jedes der vier Zielmuster werden im Folgenden Handlungsempfehlungen dargestellt und evaluiert, die auf die Erreichung dieser Zielmuster abzielen und weitestgehend auf die IT-Anwendungsintegration ausgerichtet sind. Die aufgeführten Handlungsempfehlungen basieren zum einen auf veröffentlichten Best Practices, zum anderen stammen sie aus der Unternehmens- bzw. Beratungspraxis des Autoren-Teams.

3.1 Optimierung mit Sourcing-Strategien

Sourcing-Strategien beziehen sich auf die Erbringung von Leistungen. In der betrieblichen Praxis müssen in Unternehmen oftmals Entscheidungen darüber getroffen werden, welche Leistungen intern erbracht und welche extern beschaffen werden sollen. Die Kosten zur Erbringung der Leistungen, deren Qualität, als auch das im Unternehmen vorhandene Knowhow, spielen bei solchen Entscheidungen oftmals eine wichtige Rolle [Ke17].

In Bezug auf die IT-Anwendungsintegration gilt die Knappheit der auf dem Markt verfügbaren Arbeitskräfte, die relevante Erfahrungen und Fähigkeiten vorweisen können, als große Herausforderung für Unternehmen. Das bezieht sich nicht nur auf technische Fähigkeiten, sondern auch auf das für die IT-Anwendungsintegration notwendige Wissen über die fachlichen Zusammenhänge [SA12], [Wo18]. Bei der Erarbeitung der Zielarchitektur für die IT-Anwendungsintegration sollten daher folgende Integrationsszenarien geprüft werden:

- *Externe Beschaffung von Business-to-Government (B2G)-Schnittstellen*: Dies bezeichnet Schnittstellen zwischen Unternehmen und Behörden, z.B. für die elektronische Voranmeldung von Mehrwertsteuer oder für die elektronische Ausfuhranmeldung. Derartige Schnittstellen zeichnen sich durch eine hohe Dynamik und vielfältige länderspezifische Ausprägungen aus [Li13], [MW16]. Abhängig von der Anzahl der Länder, in denen ein globales Unternehmen vertreten ist, kann dies einen enormen Entwicklungs- und Pflegeaufwand bedeuten. Eine externe Beschaffung kann die intern verfügbaren IT-Ressourcen entlasten.
- *Anbindung von externen Unternehmen über Value-Added Networks (VANs)*: Globale Unternehmen haben entlang der Supply Chain eine große Anzahl an Schnittstellen zur Kundschaft und zu liefernden Unternehmen, welche durch IT-Integration und Automatisierung zu wichtigen Wettbewerbsvorteilen führen kann. Statt eine direkte Verbindung zu jedem externen Unternehmen einzurichten, kann dies über ein Value-Added Network[2] ermöglicht werden. Somit muss das Unternehmen eine einzige Verbindung zum VAN einrichten, dessen betreibendes Unternehmen die Konfiguration und die Pflege der Verbindungen zu den externen Unternehmen dann als Dienstleistung erbringt. Die Verhandlungsmacht der beteiligten Parteien kann hier eine entscheidende Rolle spielen, d.h. dieser Ansatz ist bei Verbindungen zu liefernden Unternehmen besser und bei der Kundschaft eventuell nur bedingt durchsetzbar.
- *Einsatz vorgefertigter Adapter und Integrationspakete*: Im Falle technisch anspruchsvoller Integrationsanforderungen bieten Softwarehäuser oftmals vorgefertigte Adapter und Integrationspakete an, die in die bestehende EAI-Lösung integriert werden können. Dies gilt beispielsweise für die Anbindung von stark verbreiteten und führenden IT-Lösungen wie das CRM-System von Salesforce oder die Cloud-Plattform Amazon Web Services, da sich dabei eine große Anzahl an Unternehmen ähnlichen Herausforderungen stellen muss. Im Zuge der Erarbeitung der Zielarchitektur sollte geprüft werden, ob solche Adapter bzw. Integrationspakete in den bestehenden Lizenzen enthalten sind oder ob diese zugekauft werden können. Abhängig von den ausgehandelten Vertragskonditionen kann sich dadurch ein Unternehmen auch gegenüber potenziellen Änderungen schützen, indem das Softwarehaus zur Bereitstellung von neuen Versionen verpflichtet wird. Dies kann insbesondere hilfreich sein, wenn internes Knowhow nicht verfügbar ist, erst aufgebaut werden muss oder aktuell Kapazitätsengpässe vorliegen.

[2] Ein VAN kann in diesem Zusammenhang als eine Software as a Service (SaaS)-Plattform verstanden werden, die von einem anderen Unternehmen betrieben wird und Integrationsdienstleistungen zur Verfügung stellt.

3.2 Verbesserung Time-to-Market

Heutzutage stellt die Geschwindigkeit bei der Bewertung von Anforderungen und ihrer Umsetzung ein wichtiges Ziel der Unternehmens-IT dar [Ke17], [Ni18]. Die Zeitspanne von der ersten Idee bis zur Produktivsetzung muss möglichst kurz sein. Dies gilt auch für die Integration von IT-Anwendungen im Unternehmen. Die nachfolgenden Handlungsempfehlungen für die Zielarchitektur in Bezug auf die IT-Anwendungsintegration können die Time-to-Market verbessern:

- *Erzeugung von Skaleneffekten durch Application Programming Interfaces (APIs)*: Wiederkehrende Integrationsanforderungen können mithilfe von APIs als Services implementiert und zur Verfügung gestellt werden, um somit den internen Implementierungsaufwand möglichst gering zu halten. Dieser serviceorientierte Ansatz kann sowohl für die Integration von internen als auch von externen IT-Anwendungen verwendet werden. Im zweiten Fall kann der interne Aufwand weiterhin reduziert werden, indem die APIs um integrierte Dokumentationen, Beispielnachrichten und Testszenarien erweitert werden. Die Entscheidung über die zu entwickelnden APIs muss unternehmensspezifisch getroffen werden. Die bereits im vorherigen Abschnitt erwähnte Verhandlungsmacht der beteiligten Parteien kann hier ebenfalls eine wichtige Rolle spielen. Geeignete Einsatzszenarien könnten dementsprechend Schnittstellen zu IT-Anwendungen von liefernden oder dienstleistenden Unternehmen wie E-Procurement-Systeme, Transport Management Systeme oder Warehouse Management Systeme sein.

- *Entkopplung von Geschäfts- und Integrationslogik*: Durch die lose Kopplung von IT-Anwendungen kann die Wiederverwendbarkeit von bestimmten Schnittstellen-Komponenten (Adapter, Mappings) gefördert und somit Synergieeffekte erzielt werden. Wenn z.B. für eine Niederlassung ein spezielles Mapping für die Übertragung von Zahlungsinformationen an einer Bank implementiert wird, sollten die dafür benötigten organisationale Daten wie Buchungskreise, Kostenstellen oder Bankkonten entweder außerhalb des Mappings konfigurierbar sein oder dynamisch aus den Nachrichten entnommen werden. Dieses Mapping kann dann bei Bedarf auch für andere Geschäftsbereiche oder Standorte des Unternehmens mit geringem Aufwand konfiguriert und eingesetzt werden.

- *Schaffung von EAI-Richtlinien*: Die hohe Anzahl an zur Verfügung stehenden Integrationstechnologien und -ansätzen führt oftmals schon vor der Umsetzung neuer Anforderungen zu zeitaufwendigen Diskussionen und mehreren Abstimmungsrunden innerhalb eines EAI-Teams[3]. Die Entscheidungsfindung kann beschleunigt werden, indem Architektur- und EAI-Team Richtlinien vereinbaren, wie wiederkehrende Integrationsanforderungen zu implementieren sind. Die folgenden Punkte werden als Orientierung empfohlen:
    - Point-to-Point-Verbindungen sind bei globalen Unternehmen zu vermeiden. Hub & Spoke- oder serviceorientierte Architekturen können zur Komplexitätsreduktion beitragen[4].

[3] Damit sind die Personen gemeint, welche das entsprechende Fachwissen zur IT-Anwendungsintegration aufweisen und damit im Unternehmen betraut sind.

[4] Beide Architekturen haben Vorteile und es ist nicht zielführend im Rahmen dieses Beitrags diese miteinander zu vergleichen. Eine solche Entscheidung muss unternehmensspezifisch geprägt werden. Auch eine Kombination der beiden Ansätze ist möglich, beispielsweise durch einen Hub & Spoke-Ansatz im ersten Schritt die Point-to-Point-Verbindungen eliminieren, und darüber hinaus bestimmte Schnittstellen als Services zur Verfügung stellen, wie bei der ersten Handlungsempfehlung dieses Abschnitts beschrieben.

- Bei der Integration von On-Premise-Anwendungen werden traditionell On-Premise-Integrationssysteme eingesetzt.
- Bei der Integration von Cloud-Anwendungen, IoT-Geräten und auch bei hybriden Anbindungen, bei denen sowohl Cloud- als auch On-Premise-Anwendungen zu integrieren sind, macht es hingegen wenig Sinn, diese über ein On-Premise-Integrationssystem zu verbinden. Aktuell werden für derartige Integrationsszenarien cloudbasierte Integrationssysteme empfohlen, welche auch als Integration Platform as a Service (iPaaS) bezeichnet werden [EWK17], [KB20], [ZZL21].

- *Dokumentation mit Übersicht*: Die Time-to-Market bezieht sich nicht nur auf die Entwicklung von neuen Schnittstellen, sondern auch auf die Durchführung von Änderungen an bestehenden Schnittstellen. Dabei ist eine gute Übersicht über die entwickelten Schnittstellen bzw. Schnittstellen-Komponenten zielführend. Eine solche Dokumentation kann z.B. in einem EAM-Werkzeug erfolgen. Sie sollte die folgenden Aspekte berücksichtigen:
  - Zu den angebundenen internen und externen IT-Anwendungen werden vereinbarte technische Aspekte dokumentiert, wie z.B. eingesetztes Kommunikationsprotokoll, synchrone oder asynchrone Kommunikation sowie Richtung der Übertragung. Sind diese Aspekte toolgestützt dokumentiert, so erlaubt dies eine schnelle Identifikation der von bestimmten Änderungen betroffenen IT-Anwendungen, beispielsweise bei einer Versionsänderung eines bestimmten Kommunikationsprotokolls.
  - Wenn bei bestimmten Schnittstellen eine Transformation von Nachrichten stattfindet, sollten für eine bessere Nachvollziehbarkeit die Mappings und deren Änderungshistorie dokumentiert werden. Dafür sollte eine Vorlage entwickelt und als Standard verwendet werden. Gegebenenfalls kann die Dokumentation automatisiert über die EAI-Lösung oder über ein Add-On erstellt werden.

3.3 Reduktion von Heterogenität

Heterogenität bedeutet im Kontext einer Zielarchitektur, dass redundante technische Lösungen für ein und dieselbe Anforderung existieren, was zu einem Komplexitätszuwachs führt. Außerdem führt es zu zusätzlichen Kosten [Ke17]. Folgende Handlungsempfehlungen können der Heterogenität in Bezug auf die IT-Anwendungsintegration systematisch entgegenwirken:

- *Wiederverwendbarkeit bereits existierender Schnittstellen-Komponenten sicherstellen*: Wie schon zuvor beschrieben, können wiederverwendbare Komponenten durch die Entkopplung von Geschäfts- und Integrationslogik entstehen. Ein wichtige Voraussetzung dafür bleibt aber auch, mithilfe einer angemessenen Dokumentation die dafür erforderliche Wissensbasis zu schaffen. Auf Basis dieser Dokumentation können Entscheidungsvorlagen erarbeitet werden. Um in der Praxis Heterogenität tatsächlich zu reduzieren, müssen die entsprechenden Entscheidungen getroffen werden. Hierbei ist man häufig im Spannungsfeld lokal vs. global [ADS21]. Dies ist z.B. der Fall, wenn der Fachbereich eine schnelle, pragmatische Lösung ohne Abstimmung mit weiteren Abteilungen favorisiert, wohingegen aus der unternehmensweiten Betrachtung heraus eine über alle betroffenen Fachbereiche hinweg abgestimmte Lösung sinnvoller wäre.

- *Installation eines Technologiemanagements*: Abhängig von den spezifischen Anforderungen jedes Unternehmens sollten die EAI-Komponenten bestimmt werden, welche bestmöglich zur IT-Anwendungsintegration beitragen. Dabei liefert zum einen die IT-Strategie des Unternehmens einen wichtigen Input [WAE17], zum anderen die aktuell bestehenden Handlungsbedarfe. Das Technologieportfolio sollte dabei auf möglichst wenigen Technologien basieren [Kh15]. Weiterhin können bei der Technologieauswahl Anforderungen der

Kundschaft, Produkt-Roadmaps von Softwarehäusern oder aktuelle Entwicklungen in der IT berücksichtigt werden. Hierzu zählt beispielsweise die Tendenz, dass immer mehr IT-Anwendungen ausschließlich als SaaS angeboten werden. Dies könnte dazu führen, dass mittel- bis langfristig keine On-Premise-Integrationssysteme mehr benötigt werden, sondern nur noch iPaaS.

- *Management von Redundanz:* In manchen Fällen wird Redundanz bewusst in Kauf genommen, um höher priorisierte Ziele zu erfüllen, beispielsweise eine verbesserte Time-to-Market, um somit die Zufriedenheit der Kundschaft zu erhöhen [Ke17]). Eine unternehmensspezifische Priorisierung der Zielmuster erscheint in solchen Fällen also zweckmäßig, bedarf aber eines gesteuerten Managements der resultierenden technischen Schulden [Li20].

3.4 Bewältigung von Fusionen

Die Bewältigung von Fusionen ist ein sehr komplexes und weitreichendes Thema, das hohe Anforderungen an die Unternehmens-IT und die IT-Anwendungsintegration im Speziellen stellt [Ke17]. Die folgenden, bereits oben im Detail vorgestellten Handlungsempfehlungen zur Zielarchitektur können zur Vermeidung von Redundanzen und Komplexität im Fall einer Fusion beitragen:

- *Wiederverwendbarkeit bereits existierender Schnittstellen-Komponenten*
- *Entkopplung von Geschäfts- und Integrationslogik*
- *Dokumentation mit Übersicht*

Diese Empfehlungen bilden eine wichtige Grundlage, um zum einen bestehende Schnittstellen bzw. Schnittstellen-Komponenten im zu integrierenden Unternehmen bedarfsgerecht einzusetzen. Zum anderen kann dadurch die Integration von IT-Anwendungen zwischen den zwei Unternehmen erleichtert werden. Allerdings muss zuerst untersucht werden, wann und inwieweit eine Konsolidierung der IT-Landschaften beider Unternehmen stattfinden wird und welche IT-Anwendungen davon betroffen sind. Es würde wenig Sinn ergeben, neue Schnittstellen zu IT-Anwendungen einzurichten, welche dann im Rahmen des Fusionsprojektes abgelöst werden.

3.5 Einbettung der Handlungsempfehlungen in die EAM-Aktivitäten

Um die vorgestellten Handlungsempfehlungen in ein EAM-Vorgehen zu integrieren, müssen ausgehend von der Ist-Architektur (deren Existenz zu dem hier gewählten Zeitpunkt vorausgesetzt wird) zuerst die Zielmuster ausgewählt und priorisiert werden. Pro Zielmuster können dann die Handlungsempfehlungen herangezogen und unter Berücksichtigung des unternehmensspezifischen Kontexts diskutiert werden. Mit den daraus gewonnenen konkreten Lösungsansätzen kann mit der Erstellung der Zielarchitektur begonnen werden (s. Abbildung 1).

Um die Zielarchitektur für die IT-Anwendungsintegration umzusetzen, sind weitere Schritte nötig. Das EAM-Team unterstützt dabei die folgenden Aufgaben [Ha13], [Ke17], [Th18]: Nachdem die Zielarchitektur inklusive Zieldatum festgelegt ist, wird die Ist-Architektur erfasst und Unterschiede bzw. Lücken zur Zielarchitektur analysiert. Dies schafft Transparenz über aktuelle Problemfelder, funktionale Redundanzen, sowie Konsolidierungs- und Optimierungspotenziale. Auf dieser Basis werden Maßnahmen zur Erreichung der Zielarchitektur definiert und sinnvoll gebündelt. Dabei sind konträre Ziele bzw. Abhängigkeiten zwischen einzelnen Maßnahmen und Risiken zu berücksichtigen.

Die Erreichung der Zielarchitektur ist i.d.R. ein Vorhaben mit einem mittel- bis langfristigen Umsetzungshorizont. Die Umsetzung der Zielarchitektur erfolgt daher meist in mehreren Iterationen. In jeder dieser Iterationen wird durch die Umsetzung eines Maßnahmenbündels ein

neuer Zwischenstatus erreicht. Dieser stellt wiederum eine neue Ist-Architektur dar. Zu diesem Zeitpunkt sollte die Erreichung der durch die Iteration angestrebten Ziele kontrolliert werden. Bei Abweichungen können Korrekturmaßnahmen eingeleitet werden. Weiterhin sollte kontrolliert werden, ob die definierte Zielarchitektur und die anstehenden Maßnahmen immer noch den unternehmensspezifischen Zielen entsprechen. Zwischenzeitlich können sich Rahmenbedingungen (z.B. externe Einflussfaktoren, IT-Strategie) oder zugrundeliegende Prämissen verändert haben. In diesem Fall muss die Zielarchitektur adaptiert werden.

Abbildung 1: Prozessuale Einbettung der Handlungsempfehlungen

Für die systematische Umsetzung der Zielarchitektur spielt die EAM-Governance eine wichtige Rolle (z.B. [Ha13]). Sie sorgt u.a. dafür, dass die Umsetzung der Roadmap gesteuert und kontinuierlich überwacht wird. Sie prüft u.a. die Einhaltung der Vorgaben für die Zielarchitektur, z.B. hinsichtlich der eingesetzten technologischen Lösungen und fordert die angemessene Dokumentation der entstehenden Architektur.

4 Fallstudie und Evaluation der Ergebnisse

Die vorgestellten Handlungsempfehlungen wurden in einer Fallstudie herangezogen, die die Erstellung einer Zielarchitektur für die IT-Anwendungsintegration eines globalen Unternehmens zum Ziel hatte. Die Motivation für diese Architekturarbeiten war die fehlende strategische Ausrichtung der IT-Anwendungsintegration.

Das betrachtete Unternehmen[5], die Meier GmbH, ist schwerpunktmäßig in den Branchen Automobil und Maschinenbau tätig und beschäftigt weltweit über 20.000 Personen an rund 80 Standorten. Im Rahmen der Fallstudie wurden mit dem Unternehmen drei Workshops geplant und durchgeführt. Die Integration von IT-Anwendungen verantworten in dem Unternehmen Personen aus den beiden zentralen IT-Abteilungen „Application & Process Support“ und „Infrastructure“, welche das EAI-Team bilden. Demzufolge waren leitende Personen aus beiden Abteilungen involviert.

Im Anschluss wurden die Ergebnisse der Fallstudie mithilfe eines Applicability Checks nach Rosemann und Vessey evaluiert [RV08]. Ein Applicability Check ist eine empirische Evaluationsmethode mit welcher Theorien, Konzepte, Modelle und Artefakte hinsichtlich ihrer Wichtigkeit, Relevanz und Eignung für die Lösung eines praxisrelevanten Problems bewertet werden

[5] Zur Gewährleistung der Anonymität wird der Unternehmensname pseudonymisiert. Auch weitere Informationen (z.B. die Geschäftsbereiche) werden anonymisiert dargestellt.

können. Die Bewertung erfolgt dabei durch Personen aus der Praxis, die über das erforderliche Fachwissen verfügen [RV08].

4.1 Unternehmenskontext und Ist-Architektur

Den Kern des Unternehmens bilden zwei ERP-Systeme: Geschäftsbereich A, der im Moment als wichtigster Geschäftsberiech verstanden werden kann, hat das On-Premise SAP ERP Central Component (ECC) seit über 20 Jahren global im Einsatz. Der jüngere Geschäftsbereich B führte von Anfang an ein cloudbasiertes ERP-System ein, nämlich S/4HANA Cloud. Darüber hinaus gibt es weitere IT-Anwendungen als On-Premise-Installationen oder in der Cloud.

Die existierende Integrationsstrategie ist mittlerweile rund 10 Jahre alt. Sie zielte auf die Systematisierung und zukünftige Ausrichtung der IT-Anwendungsintegration ab. Jedoch ist sie schwerpunktmäßig technisch orientiert: Abhängig von den Integrationsanforderungen legt sie fest, welches der zwei Integrationssysteme, der Axway TradeSync Integration Manager (TSIM) und das damals neue Integrationssystem SAP Process Integration (PI), eingesetzt werden soll. Beide Integrationssysteme boten unterschiedliche technische Möglichkeiten. Daher orientierte sich die Entscheidung an den jeweils benötigten Kommunikationsprotokollen und Formaten bzw. Standards zur IT-Anwendungsintegration. Mittlerweile überschneiden sich beide Integrationssysteme – welche heute noch als On-Premise-Installationen im Einsatz sind – in vielen Funktionalitäten. Zusätzlich gibt es seit ein paar Jahren ein cloudbasiertes Integrationssystem, die SAP Cloud Platform Integration (CPI), worüber jedoch momentan nur wenige Integrationsszenarien abgewickelt werden. Abbildung 2 zeigt für die beiden Geschäftsbereiche die aktuelle Nutzung dieser Integrationssysteme.

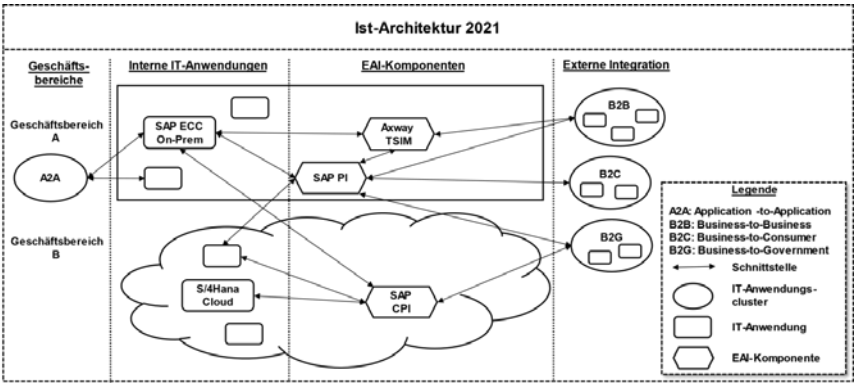


Abbildung 2: Ist-Architektur der Meier GmbH in Bezug auf die IT-Anwendungsintegration im Jahr 2021. Quelle: Eigene Darstellung

Da die Integrationsstrategie seit mehreren Jahren nicht mehr aktualisiert wurde, berücksichtigt sie weder die Cloud-Thematik noch andere aktuelle Herausforderungen der IT-Anwendungsintegration. Dementsprechend gilt sie als veraltet und ungültig. Dies hat zur Folge, dass die Auswahl des zu verwendenden Integrationssystems und -ansatzes oftmals auf willkürlichen Entscheidungen oder auf den jeweils zur Verfügung stehenden personellen Ressourcen basiert. Dies führt zu Inkonsistenzen und Redundanzen.

Die Analyse der Ist-Architektur für die IT-Anwendungsintegration zeigt verschiedene positive Aspekte, die es beizubehalten gilt, wie die Entkopplung von Geschäfts- und Integrationslogik und das Vorhandensein von Dokumentationen bei den meisten Mappings. Handlungsbedarf gibt es hingegen bzgl. der funktionalen Redundanzen zwischen den Integrationssystemen und der Inkonsistenzen bei der Umsetzung von B2G- und hybriden Schnittstellen.

Als Zieldatum für die Erreichung der Zielarchitektur hat die Meier GmbH das Jahr 2027 festgelegt, aufgrund von wichtigen Ereignissen, die einen direkten Einfluss auf die IT-Anwendungsintegration haben werden:

- Das Integrationssystem SAP PI wird ab 2027 nicht mehr unterstützt. Dies beruht auf der aktuellen Produkt-Roadmap des Softwarehauses SAP SE.
- Das Format Intermediate Document (IDoc), welches momentan das meistverwendete Nachrichtenformat im Unternehmen darstellt, wird ab 2027 nicht bzw. nicht in der gleichen Form existieren. Besonders betroffen sind davon Schnittstellen bzw. Mappings zwischen dem Geschäftsbereich A und der Kundschaft für die Übertragung von Lieferavisen über den Axway TSIM.

Eine weitere strategische Entscheidung, welche die IT-Anwendungsintegration im Unternehmen beeinflussen wird, ist die für 2024 geplante Einführung des neuen ERP-Systems SAP S/4HANA für den Geschäftsbereich A, welches das SAP ECC ersetzen wird. Dies wird voraussichtlich weiterhin eine On-Premise-Installation sein. Diese strategischen Entscheidungen betreffen ausschließlich den Geschäftsbereich A (siehe Abbildung 3).

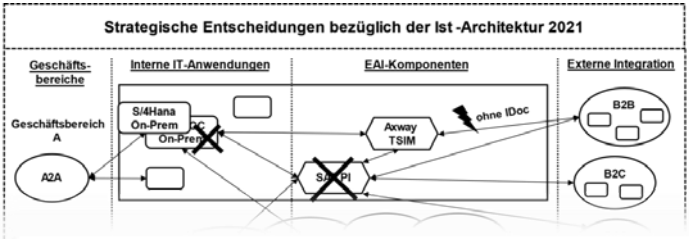


Abbildung 3: Strategische Entscheidungen zur Ist-Architektur. Quelle: Eigene Darstellung.

Die Erarbeitung der Zielarchitektur erfolgte in der Fallstudie anhand des in Abbildung 1 skizzierten Vorgehens.

4.2 Auswahl und Priorisierung der relevanten Zielmuster

Die genannten Rahmenbedingungen sowie die strategischen Entscheidungen in Bezug auf zentrale IT-Anwendungen haben großen Einfluss auf die Zielarchitektur für die IT-Anwendungsintegration. Für die Erarbeitung der Zielarchitektur wurden im ersten Schritt die für die Meier GmbH relevanten Zielmuster ausgewählt und priorisiert. In diesem Fall sind außer der *Bewältigung von Fusionen* alle Zielmuster relevant. Höchste Priorität hat die *Reduktion von Heterogenität*, eine Herausforderung, die bereits bei der Betrachtung der Ist-Architektur mit den heterogenen Ausprägungen für Geschäftsbereich A und B klar erkennbar ist. Die *Verbesserung der Time-to-Market* steht für die Meier GmbH an zweiter Stelle, da die Laufzeiten von IT-Projekten als zu lang empfunden werden. An dritter Stelle steht die *Optimierung mit Sourcing-Strategien*, wobei hier der Fokus auf den B2G-Schnittstellen liegt.

4.3 Diskussion der Handlungsempfehlungen

Im zweiten Schritt wurden pro Zielmuster die Handlungsempfehlungen für die Meier GmbH diskutiert. Dabei wurden die EAI-Komponenten identifiziert, die zukünftig benötigt werden, und andere, die nicht mehr eingesetzt werden können. Es wurde definiert, wie Schnittstellen zu unterschiedlichen Anwendungsclustern (A2A, B2B, B2C und B2G) zukünftig umgesetzt und welche Integrationsszenarien als APIs zur Verfügung gestellt werden sollen. Darüber hinaus

wurde eine Bewertungsmatrix entwickelt, die als gemeinsame Entscheidungsbasis für die Bewertung zukünftiger Integrationsanforderungen dienen soll.

4.3.1 Reduktion von Heterogenität

Bei der Diskussion der Handlungsempfehlungen zur Reduktion von Heterogenität, welches für die Meier GmbH das höchst priorisierte Zielmuster darstellt, wurden u.a. die IT-Strategie des Unternehmens und die Produkt-Roadmaps der relevanten Softwarehäuser berücksichtigt. Dabei wurden die folgenden beiden Zielvorgaben erarbeitet.

(1) Im Rahmen des **Technologiemanagements** wurden die benötigten EAI-Komponenten wie folgt festgelegt:

- Das Nachfolgeprodukt der SAP PI wird das cloudbasierte Integrationssystem SAP CPI sein, welches SAP SE als strategische Integrationsplattform der Zukunft ansieht. Eine Migration der Integrationsszenarien muss bis dahin erfolgen.
- Der Axway TSIM wird weiterhin bestehen. Es wird erwartet, dass das Softwarehaus Axway Software SA eine einfache Migrationsmöglichkeit für die von der SAP SE angekündigte IDoc-Abschaffung anbieten wird. Sollte dies nicht der Fall sein, müssen im Unternehmen ca. 150 der insgesamt 750 verfügbaren Mappings neu entwickelt oder angepasst werden.
- Eine API-Komponente muss zusätzlich erworben bzw. lizenziert werden.
- A2A: Für die Integration von internen On-Premise-Anwendungen scheint die SAP SE keine Ersatzlösung zu SAP PI anzubieten. Der Axway TSIM ist nur für B2B-Schnittstellen geeignet. Eine Anbindung über die SAP CPI wird als Umweg angesehen und es könnte zu Performance-Problemen führen. Hier ist die Anschaffung einer zusätzlichen EAI-Komponente zu prüfen und eine entsprechende Evaluation durchzuführen.

(2) Die **Wiederverwendbarkeit existierender Schnittstellen-Komponenten** soll weiterhin gefördert werden, insbesondere in Bezug auf die Mappings. Momentan wird das im Axway TSIM und in der SAP PI durch die Entkopplung von Geschäfts- und Integrationslogik weitestgehend erreicht. In der SAP CPI waren jedoch die ersten Versuche nicht erfolgreich. Dementsprechend ist der Aufbau von Kompetenzen (beispielsweise durch Schulungen oder Weiterbildungsmaßnahmen) für dieses Integrationssystem erforderlich.

4.3.2 Verbesserung der Time-to-Market

An zweiter Stelle steht für die Meier GmbH die Verbesserung der Time-to-Market. Zur bestmöglichen Erreichung dieses Zielmusters wurden Integrationsszenarien identifiziert, welche sich für die Entwicklung von APIs eignen. Darüber hinaus wurden EAI-Richtlinien festgelegt sowie das Thema der Dokumentation besprochen. Die Ergebnisse sind unter folgenden drei Punkten zusammengefasst:

(1) **APIs** sollen für folgende Integrationsszenarien zur Erzeugung von Skaleneffekten entwickelt und zur Verfügung gestellt werden:

- Anbindung von Unternehmen, welche Fahrerlose Transportfahrzeuge im Lager als Dienstleistung anbieten (Automated Guided Vehicle),
- Anbindung von Unternehmen, welche ein externes Lager für das Unternehmen betreiben (Third Party Logistics), und
- Anbindung von Klein- und Einzelkunden*innen, um Bestellungen automatisiert aufgeben zu können (Order Management System).

(2) Um klare **EAI-Richtlinien** zu schaffen, wurde eine Bewertungsmatrix entwickelt, die als Entscheidungsbasis für die Bewertung von Integrationsanforderungen dienen soll. Dabei wurden Kriterien zur Bewertung von Umsetzungsoptionen festgelegt: Kosten (Entwicklung, Lizen-

zen), Geschwindigkeit, Machbarkeit, Security, Wartung, Verfügbarkeit und Latenzen, also spezielle Anforderungen wie Bandbreite, Performance usw. Darüber hinaus wurden folgende EAI-Richtlinien vereinbart:

- Bei neuen Anforderungen wird die SAP CPI vor der SAP PI schon jetzt bevorzugt, wenn es technisch möglich ist. Dadurch soll der spätere Migrationsaufwand reduziert werden.
- Cloud-to-Cloud- und hybride Anbindungen erfolgen über die SAP CPI.
- Verbindungen zu externen IT-Anwendungen erfolgen nur über ein Integrationssystem und nur unter Verwendung eines Kommunikationsprotokolls, welches Verschlüsselung anbietet (Integration von IT-Security-Richtlinien).
- B2B-Schnittstellen werden in zwei Kategorien eingeteilt:
    - B2B-klassisch: Klassische B2B-Schnittstellen werden weiterhin über den Axway TSIM abgewickelt. Damit sind Schnittstellen des Geschäftsbereichs A zu externen On-Premise-Anwendungen anderer Unternehmen gemeint.
    - B2B-neu: Bei neuartigen Integrationsanforderungen (z.B. bei Verbindungen zu Cloud-Anwendungen) wird unabhängig vom Geschäftsbereich die SAP CPI eingesetzt.
- B2C-Schnittstellen werden über die CPI angebunden, da der Axway TSIM dafür nicht geeignet ist.

(3) Hinsichtlich der **Dokumentation** zeigten sich vielfältige Sichtweisen und unterschiedliche Anforderungen an Inhalt, Umfang und Form der Dokumentation. In einem ersten Schritt wurde vereinbart, dass Mappings weiterhin wie gehabt dokumentiert werden sollen. Es sind jedoch noch Maßnahmen zu ergreifen, um den Verantwortlichen die Dokumentation zu erleichtern. Damit die erstellten Übersichten mit den angebundenen IT-Anwendungen weiterhin gepflegt und aktualisiert werden, sind organisatorische Maßnahmen notwendig. Beispielsweise sollte eine dedizierte EAM-Rolle geschaffen werden.

4.3.3 Optimierung mit Sourcing-Strategien

Anhand der Handlungsempfehlungen zum Zielmuster Optimierung mit Sourcing-Strategien wurde diskutiert, welche Leistungen aufgrund von vorhandenen oder aufzubauenden Kompetenzen intern erbracht werden sollen und welche Leistungen extern hinzugekauft werden sollen. Dabei wurden die folgenden beiden Zielvorgaben festgelegt:

(1) Hinsichtlich der **Beschaffung von B2G-Schnittstellen** sind in der CPI die vorgefertigten, von der SAP SE angeboten Integrationspakete zu verwenden. Es erfolgt keine Eigenentwicklung, jedoch ist internes Knowhow für die Implementierung der Integrationspakete und die operative Unterstützung bei Updates notwendig. Aus Compliance-Gründen ist sicherzustellen, dass immer die aktuelle Version des Integrationspaketes implementiert ist. Es ist zulässig, dass bei speziellen länderspezifischen Anforderungen zusätzliche Dienstleistungen von lokalen Beratungsunternehmen bezogen werden.

(2) **VANs** sollen für die Anbindung externer Unternehmen weiterhin genutzt werden. Das ist ein Ansatz, der im Unternehmen heutzutage schon Anwendung findet.

4.4 Erstellung der Zielarchitektur

Nach Anwendung der oben beschriebenen Handlungsempfehlungen auf die Ist-Architektur wurde die Zielarchitektur erstellt (siehe Abbildung 4).

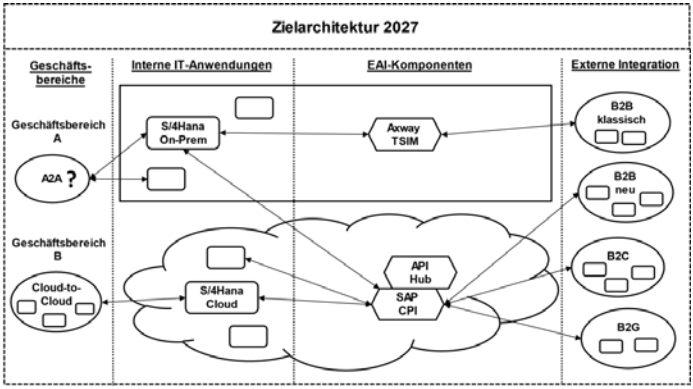


Abbildung 4: Zielarchitektur für die IT-Anwendungsintegration im Jahr 2027

Anschließend wurden Maßnahmen definiert, um die Zielarchitektur zu erreichen und die Roadmap festgelegt. Ein wichtiger Meilenstein ist das Jahr 2024, da bis dahin das neue ERP-System des Geschäftsbereichs A eingeführt sein soll. In der ersten Phase der Roadmap (2021-2024) soll daher der Fokus auf die Behebung der identifizierten Problemfelder, die Standardisierung und die Reduktion von Komplexität gelegt werden. Die erste Phase kann somit als vorbereitender Schritt verstanden werden, in dem keine großen und aufwendigen Änderungen stattfinden werden. In der zweiten Phase (2024-2027) wird es voraussichtlich größere Änderungen und Umstellungen geben, beispielsweise die Inbetriebnahme der neuen API-Komponente und die Ablösung des Integrationssystems SAP PI.

4.5 Applicability Check

Die Ergebnisse der Fallstudie wurden direkt im Anschluss an die Durchführung der Fallstudie analysiert, strukturiert und allen teilnehmenden Personen zur Verfügung gestellt. Diese wurden gebeten, einen schriftlichen Fragebogen im Zeitraum vom 25. Juli bis zum 6. August 2021 auszufüllen. Der Fragebogen beinhaltete zum einen Multiple-Choice-Fragen, zum anderen wurde auch die Möglichkeit zur Ergänzung von Anmerkungen und Kommentaren gegeben. Die Evaluation der Multiple-Choice-Fragen basiert auf dem Ansatz von Aier und Fischer, welche für diesen Zweck die folgenden Evaluationsmerkmale definieren [AF11]:

- Nützlichkeit: Das Artefakt erfüllt seinen Zweck und gleichzeitig ist dieser Zweck nützlich bzw. relevant für die das Artefakt nutzenden Personen.
- Interne Konsistenz: Die Elemente des Artefaktes sind in sich, aber auch zueinander, konsistent und widerspruchsfrei.
- Externe Konsistenz: Das Artefakt ist im Einklang mit dem allgemeinen Wissen aus der jeweiligen Disziplin und baut auf bestehenden Theorien oder Konzepten auf[6].
- (Breiter Anwendungsbereich: Dieses Kriterium entfällt hier, weil es sich auf eine unternehmensspezifische Zielarchitektur nicht anwenden lässt.)
- Einfachheit: Das Artefakt ist gut lesbar, verständlich und nicht komplexer als es sein muss.

[6] Es gibt aber auch revolutionäre Artefakte, die etablierten und allgemein akzeptierten Theorien widersprechen und über diesen Weg zu neuen Erkenntnissen führen [Ku77].

- Gewinnung neuer Erkenntnisse: Das Artefakt offenbart neue Einsichten und bietet eine Grundlage für zukünftige Forschung bzw. Weiterentwicklungen.

Die erstellte Zielarchitektur wurde in den Multiple-Choice-Fragen weitgehend positiv bewertet. Die Evaluationsmerkmale „Nützlichkeit", „externe Konsistenz" und „Gewinnung neuer Erkenntnisse" wurden von allen drei teilnehmenden Expert*innen als vollständig erfüllt angesehen. Bei den restlichen Evaluationsmerkmalen, nämlich „interne Konsistenz" und „Einfachheit", gab es jeweils zwei Stimmen für die vollständige Erfüllung und eine für die Erfüllung zum größten Teil. Dies wird in der nachfolgenden Tabelle veranschaulicht:

Antworten / Evaluationsmerkmale	Stimmt	Stimmt eher	Stimmt eher nicht	Stimmt nicht
Nützlichkeit	3	0	0	0
Interne Konsistenz	2	1	0	0
Externe Konsistenz	3	0	0	0
Einfachheit	2	1	0	0
Gewinnung neuer Erkenntnisse	3	0	0	0

Tabelle 1: Ergebnisse der Multiple-Choice-Fragen

Um die Antworten der Multiple-Choice-Fragen besser zu verstehen, werden nun die ergänzenden Anmerkungen und Kommentare zu jedem Evaluationsmerkmal zusammenfassend dargestellt:

Nützlichkeit: Die erstellte Zielarchitektur wird im Unternehmen als Grundlage zur strategischen Ausrichtung der IT-Anwendungsintegration in den nächsten Jahren angesehen. Sie schafft Transparenz und berücksichtigt aktuelle Entwicklungen im Bereich der IT-Anwendungsintegration. Jedoch betonten zwei Personen auch die Notwendigkeit zur kontinuierlichen Überprüfung, Weiterentwicklung und Anpassung.

Interne Konsistenz: Die erstellte Zielarchitektur wird als weitgehend konsistent und widerspruchsfrei wahrgenommen. Eine Person merkte an, dass in manchen Fällen die verwendeten Zielmuster konträre Auswirkungen haben können und dementsprechend immer eine unternehmensspezifische Priorisierung notwendig ist, wie sie für die Meier GmbH auch erfolgt ist. Weiterhin wurde erwähnt, dass einige der genannten Technologien noch nicht ausgereift sind. Das trifft beispielsweise auf das Integrationssystem SAP CPI zu, das zurzeit noch einen geringeren Funktionsumfang im Vergleich zur SAP PI aufweist. Demzufolge erweist sich dort die Entwicklung und das Monitoring von Schnittstellen oftmals als schwieriger. Die erarbeiteten Ansätze und Best Practices müssten daher in den nächsten Jahren kontinuierlich überprüft werden.

Externe Konsistenz: Diesbezüglich wurde erwähnt, dass die erstellte Zielarchitektur im Einklang mit theoretischem Wissen aus dem Bereich der IT-Anwendungsintegration sind und dieses mit der Praxis gut verbinden.

Einfachheit: Zwei Personen waren hier der Meinung, dass dieses Evaluationsmerkmal vollständig erfüllt wird. Eine Person davon erwähnte, dass die erstellte Zielarchitektur von unterschiedlichen Zielgruppen verstanden werden kann, sowohl auf der strategischen Ebene als auch auf der operativen Ebene.

Gewinnung neuer Erkenntnisse: Eine Aussage in diesem Zusammenhang war, dass die erstellte Zielarchitektur für die IT-Anwendungsintegration die Identifikation von Redundanz und Standardisierungspotenzial ermöglicht. Dementsprechend kann sie zur Verbesserung der Profitabilität des Unternehmens beitragen. Weiterhin wurde positiv bewertet, dass die IT-Anwendungsintegration nicht ausschließlich aus einer operativen bzw. technischen Perspektive betrachtet wird, sondern auch auf einer strategischen und etwas abstrakteren Ebene. Dadurch können die

Anforderungen, welche für die IT-Anwendungsintegration im Unternehmen relevant sind, analysiert, kategorisiert und systematisiert werden.

5 Fazit

Die Integration von IT-Anwendungen ist heutzutage ein komplexes und herausforderndes Thema für viele Unternehmen. In diesem Kontext ist vor allem die Entwicklung einer Zielarchitektur für die IT-Anwendungsintegration schwierig. Um Unternehmen diese Aufgabe zu erleichtern, hat dieser Beitrag Handlungsempfehlungen für die Erstellung einer Zielarchitektur für die IT-Anwendungsintegration entwickelt.

5.1 Zusammenfassung

Anhand einer systematischen Literaturanalyse wurde der aktuelle Stand der Forschung aus den zwei relevanten Themenbereichen EAM und EAI ermittelt. Eine zentrale Erkenntnis dabei war, dass die Verbindung von EAI und EAM bzw. die Betrachtung der IT-Anwendungsintegration aus der Perspektive des EAM als unzureichend erforscht gilt. Der Zielstellung entsprechende Handlungsempfehlungen stellen bis heute ein Forschungsdesiderat dar.

Aufbauend auf den Zielmustern von Keller wurden 13 Handlungsempfehlungen ausgesprochen. In Bezug auf das Zielmuster Optimierung mit Sourcing-Strategien wird die externe Beschaffung von B2G-Schnittstellen, die Anbindung von externen Unternehmen über VANs sowie der Einsatz von vorgefertigten Adaptern und Integrationspaketen für technisch anspruchsvolle Integrationsanforderungen empfohlen. Im Mittelpunkt des zweiten Zielmusters steht die Verbesserung der Time-to-Market. Eine Handlungsempfehlung in diesem Zusammenhang ist, den internen Implementierungsaufwand bei wiederkehrenden Integrationsanforderungen mithilfe von APIs zu reduzieren. Als besonders geeignet werden dabei Schnittstellen zu liefernden oder dienstleistenden Unternehmen angesehen. Weiterhin wird empfohlen, die Wiederverwendbarkeit von Schnittstellen-Komponenten durch die Entkopplung von Geschäfts- und Integrationslogik zu fördern. Des Weiteren soll die Geschwindigkeit bei der Bewertung von Anforderungen und ihrer Umsetzung durch die Schaffung von EAI-Richtlinien und mithilfe einer Dokumentation – welche zu einer besseren Übersicht über die vorhandenen Schnittstellen bzw. Schnittstelle-Komponenten beitragen kann – erhöht werden. Im Rahmen des dritten Zielmusters, bei dem es um die Reduktion von Heterogenität geht, gilt die Förderung von Wiederverwendbarkeit ebenso als wichtige Handlungsempfehlung. Darüber hinaus bietet sich die Installation eines Technologiemanagements an, um, abhängig von den spezifischen Anforderungen jedes Unternehmens, die EAI-Komponenten zu bestimmen, welche bestmöglich zur IT-Anwendungsintegration beitragen. Zudem wird betont, dass das erfolgreiche Management von Redundanz eine unternehmensspezifische Priorisierung der relevanten Zielmuster erfordert. Zur erfolgreichen Bewältigung von Fusionen, welche das letzte betrachtete Zielmuster darstellt, gelten einige der vorherigen Handlungsempfehlungen (die Wiederverwendbarkeit bereits existierender Schnittstellen-Komponenten, die Entkopplung von Geschäfts- und Integrationslogik und die Dokumentation mit Übersicht) als relevant. Es ist jedoch wichtig zu erwähnen, dass im Kontext einer Fusion eventuell auch andere Themen wie z.B. die Entscheidung zur Konsolidierung der IT-Landschaften der beiden Unternehmen höher priorisiert sind.

Basierend auf diesen Handlungsempfehlungen erfolgte die Erstellung einer Zielarchitektur für die IT-Anwendungsintegration im Rahmen einer Fallstudie, um deren Funktionsfähigkeit in einem existierenden globalen Unternehmen praktisch zu demonstrieren. Abschließend wurden die Ergebnisse der Fallstudie durch einen Applicability Check evaluiert. Somit konnte der Mehrwert der erarbeiteten Handlungsempfehlungen hervorgehoben und der Erreichungsgrad der Zielsetzung dieses Beitrags dargestellt werden.

5.2 Kritische Betrachtung und Ausblick

Die erarbeiten Handlungsempfehlungen unterstützen Unternehmen, die eine Vielzahl an zu integrierenden internen und externen IT-Anwendungen im Einsatz haben, bei der Entwicklung einer Zielarchitektur für die IT-Anwendungsintegration. Sie sind daher insbesondere für große, ggf. auch global tätige Unternehmen relevant, wie auch in der Fallstudie dargestellt. Einzelne Handlungsempfehlungen können – wenn entsprechende Integrationsanforderungen gegeben sind – gegebenenfalls auch für kleine bis mittelständische Unternehmen relevant sein.

Ein Kritikpunkt an dem Beitrag könnte die Übernahme der Zielmuster aus dem EAM-Ansatz von Keller sein. Dies wurde dadurch begründet, dass sich diese optimal zur unternehmensspezifischen Anpassung einer Zielarchitektur eignen, indem die Auswahl und die Priorisierung abhängig von den Zielen des jeweiligen Unternehmens erfolgen kann. Sicherlich gibt es auch andere Möglichkeiten, um Zielmuster für eine IT-Zielarchitektur zu definieren, beispielsweise durch eine Befragung von Personen mit Fachwissen oder durch eine vergleichende Betrachtung mehrerer EAM-Ansätze. Dies war jedoch nicht der Schwerpunkt dieses Beitrags und wie auch von Helmes und Hauer betont, muss im Rahmen des EAM das Rad nicht jedes Mal neu erfunden werden [HH20].

Die Validierung der Fallstudienergebnisse erfolgte mittels eines Applicability Checks, um ihren Zielerreichungsgrad objektiv zu bewerten. Dies ist weitgehend gelungen. Jedoch weisen alle teilnehmenden Personen eine lange Betriebszugehörigkeit auf und somit ist eine gewisse Betriebsblindheit nicht auszuschließen. Einige Aussagen sollten dementsprechend kritisch betrachtet werden. Weiterhin ist eine Bewertung von nur drei Personen alles andere als repräsentativ. Es handelt sich jedoch um diejenigen, die im betrachteten Unternehmen die IT-Anwendungsintegration auf dieser strategischen und taktischen Ebene verantworten. Zusätzlich muss erwähnt werden, dass sich die Validierung auf die konkreten Ergebnisse der Fallstudie bezieht, nämlich auf die Zielarchitektur für die IT-Anwendungsintegration bei der Meier GmbH. Daher sind die Evaluationsergebnisse nur indirekt auf die Qualität der Handlungsempfehlungen übertragbar. Mit Blick auf die Zukunft wäre es dementsprechend hilfreich, diese Handlungsempfehlungen im Rahmen weiterer Fallstudien bei anderen Unternehmen einzusetzen und somit aussagekräftigere Evaluationsergebnisse zu bekommen.

Im Rahmen dieses Beitrags wurde bei der Betrachtung der IT-Anwendungsintegration der Fokus primär auf die IT-Anwendungsarchitektur gerichtet. Zwar wurde auch die Geschäftsarchitektur auf Strategie- und Organisationsebene betrachtet, etwa indem die Ziele, die Prioritären und die IT-Strategie des Unternehmens als Input herangezogen wurden. Nachfolgende Forschungsarbeiten könnten hier aber wirksam anknüpfen und weitere Elemente der Geschäftsarchitektur heranziehen. Auch die IT-Infrastrukturebene kann wichtige Erkenntnisse in Bezug auf die IT-Anwendungsintegration liefern. Zum Beispiel könnten Redundanzen und Konsolidierungspotenziale durch die Analyse der Geschäftsprozesse identifiziert werden, welche die zu integrierenden IT-Anwendungen unterstützen. Auch die Betrachtung der Hardware-, Infrastruktur- und Netzwerkkomponenten, welche bei der IT-Anwendungsintegration zum Einsatz kommen, kann wichtige Hinweise geben.

Literaturverzeichnis

[ADS21] Aier, S.; Dinter, B.; Schelp, J.: Management of Enterprise-Wide Information Systems. In (Aier, S.; Rohner, P.; Schelp, J. Hrsg.): Engineering the Transformation of the Enterprise. A Design Science Research Perspective. Springer International Publishing, Cham, S. 201–224, 2021.

[AF11] Aier, S.; Fischer, C.: Criteria of progress for information systems design theories. Information Systems and e-Business Management 1/9, S. 133–172, 2011.

[AS05] Aier, S.; Schönherr, M.: EAI als integrierendes Element einer nachhaltigen Unternehmensarchitektur. In (Aier, S.; Schönherr, M. Hrsg.): Unternehmensarchitekturen und Systemintegration. GITO, Berlin, S. 3–56, 2005.

[Be14] Bente, S.: Kollaborative Enterprise-Architektur – Managementwerkzeug für komplexe IT-Systeme. In (Schoeneberg, K.-P. Hrsg.): Komplexitätsmanagement in Unternehmen. Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern. Springer Fachmedien Wiesbaden, Wiesbaden, S. 187–223, 2014.

[EWK17] Ebert, N.; Weber, K.; Koruna, S.: Integration Platform as a Service. Business & Information Systems Engineering 5/59, S. 375–379, 2017.

[Ha13] Hanschke, I.: Strategisches Management der IT-Landschaft. Ein praktischer Leitfaden für das Enterprise Architecture Management. Hanser, München, 2013.

[HH20] Helmes, A.; Hauer, D.: Das Zusammenspiel zwischen TOGAF®, ArchiMate® und EA-Szenarien. In (Karagiannis, D.; Moser, C.; Helmes, A. Hrsg.): Benutzerzentrierte Unternehmensarchitekturen. Ein portfolio-orientierter Ansatz zur Geschäftstransformation mit ArchiMate®. Springer Fachmedien Wiesbaden, Wiesbaden, S. 32–58, 2020.

[HRZ21] Huber, M.; Rentrop, C.; Zimmermann, S.: IT-Integration in Zeiten von Digitalisierung – (k)ein alter Hut? HMD Praxis der Wirtschaftsinformatik 2/58, S. 425–443, 2021.

[HW03] Hohpe, G.; Woolf, B.: Enterprise Integration Patterns. Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, Boston, MA, 2003.

[IB20] IBM: Application Integration. https://www.ibm.com/cloud/learn/application-integration, Stand: 30.01.2022.

[Ju21] Junker, A.: Integration Architectures in a Microservice World. https://conferences.isaqb.org/software-architecture-gathering/integration-architectures-in-a-microservice-world/, Stand: 20.01.2022.

[KB20] Kumar, A.; Bawa, S.: DAIS: dynamic access and integration services framework for cloud-oriented storage systems. Cluster Computing 4/23, S. 3289–3308, 2020.

[Ke17] Keller, W.: IT-Unternehmensarchitektur. Von der Geschäftsstrategie zur optimalen IT-Unterstützung. dpunkt.verlag, Heidelberg, 2017.

[Kh15] Khosroshahi, P. A.; Hauder, M.; Schneider, A. W.; Matthes, F.: Enterprise Architecture Management Pattern Catalog. Version 2.0. November 2015. https://wwwmatthes.in.tum.de/file/650rskbruex4/Sebis-Public-Website/Research/Enterprise-Architecture-Management/Enterprise-Architecture-Management-Pattern-Catalog-V2-2015-/EAMPC-V2-Final-Report/eampc2015.pdf, Stand: 30.01.2022.

[Ku77] Kuhn, T. S.: The Essential Tension. Selected Studies in Scientific Tradition and Change. The University of Chicago Press, Chicago, IL, 1977.

[Li13] Li, S.-H. et al.: Business-to-government application integration framework: A case study of the high technology industry in Taiwan. Computer Standards & Interfaces 6/35, S. 582–595, 2013.

[Li20] Lilienthal, C.: Langlebige Software-Architekturen. Technische Schulden analysieren, begrenzen und abbauen. dpunkt.verlag, Heidelberg, 2020.

[LU13] Lässig, J.; Ullrich, M.: Enterprise Application Integration -The Cloud Perspective. In (Daniel, F.; Dolog, P.; Li, Q. Hrsg.): Web Engineering. ICWE 2013. Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 522–525, 2013.

[MB18] Mangiapane, M.; Bender, M.: Enterprise Architekturmanagement (EAM) in einem Krankenhaus. HMD Praxis der Wirtschaftsinformatik 5/55, S. 1022–1047, 2018.

[MB20] Mangiapane, M.; Bender, M.: Patientenorientierte Digitalisierung im Krankenhaus. IT-Architekturmanagement am Behandlungspfad. Springer Vieweg, Wiesbaden, 2020.

[Me13] Mertens, P.: Integrierte Informationsverarbeitung 1. Operative Systeme in der Industrie. Springer Fachmedien Wiesbaden, Wiesbaden, 2013.

[Mü05] Müller, J.: Workflow-based Integration. Grundlagen, Technologien, Management. Springer, Berlin, Heidelberg, 2005.

[MW16] Mondorf, A.; Wimmer, M. A.: Requirements for an Architecture Framework for Pan-European E-Government Services. In (Scholl, H. J. et al. Hrsg.): Electronic Government. 15th IFIP WG 8.5 International Conference, EGOV 2016, Guimarães, Portugal, September 5-8, 2016, Proceedings. Springer, Cham, Berlin, Heidelberg, S. 135–150, 2016.

[Ni18] Niemann, K. D.: Unternehmensarchitektur und Digitalisierung. HMD Praxis der Wirtschaftsinformatik 5/55, S. 907–927, 2018.

[RMR17] Ritter, D.; May, N.; Rinderle-Ma, S.: Patterns for emerging application integration scenarios: A survey. Information Systems 67, S. 36–57, 2017.

[RV08] Rosemann; Vessey: Toward Improving the Relevance of Information Systems Research to Practice: The Role of Applicability Checks. MIS Quarterly 1/32, S. 1–22, 2008.

[SA12] Soomro, T. R.; Awan, A. H.: Challenges and Future of Enterprise Application Integration. International Journal of Computer Applications 7/42, S. 42–45, 2012.

[Th18] The Open Group Architecture Forum: TOGAF® Standard, Version 9.2, 2018.

[Ti11] Tiemeyer, E.: Enterprise Architecture Management (EAM) – IT-Architekturen erfolgreich planen und steuern. In (Tiemeyer, E. Hrsg.): Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis. Hanser, München, S. 85–138, 2011.

[WAE17] Welge, M. K.; Al-Laham, A.; Eulerich, M.: Strategisches Management. Grundlagen – Prozess – Implementierung. Springer Fachmedien Wiesbaden, Wiesbaden, 2017.

[Wo18] Wong, J.: Enterprise Application Integration. In (Liu, L.; Özsu, M. T. Hrsg.): Encyclopedia of Database Systems. Springer New York, New York, NY, S. 1301–1307, 2018.

[WW02] Webster, J.; Watson, R. T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly 2/26, S. XIII–XXIII, 2002.

[ZZL21] Zhao, Y.; Zhang, Z.; Li, J.: A Secure Edge-Cloud Computing Framework for IoT Applications. In (Lin, Y.-B.; Deng, D.-J. Hrsg.): Smart Grid and Internet of Things. 4th EAI International Conference, SGIoT 2020, TaiChung, Taiwan, December 5–6, 2020, Proceedings. Springer, Cham, Berlin, Heidelberg, S. 70–78, 2021.

Calculating the Test Costs of Micro Services in Agile Development Projects

Harry M. Sneed
ZTP-Prentner Digital, Vienna, Austria

1 Introduction to testing Micro Services

The testing approach presented here is to fit the testing effort required to test a web service with the time allowed in an agile project and still attain a minimum test coverage. Agile projects are governed by a release cycle of maximum 4 weeks [Beck01]. The project reported on here is such an agile scrum project with a tight release schedule. Every sprint should deliver a new release, i.e., a new micro service with a test coverage of at least 90% branch = C1 coverage. Past experience has shown that testing requires at least 50% of the total effort to deliver a new service [Black99]. Of that, one half of the effort is for unit testing and the other half for integration testing. Thus, if the goal is to produce a new micro service every 4 weeks, then two weeks will be for design and coding and two weeks for testing. Even if the code of a service is taken over from existing code, i.e. stripped and wrapped, it still has to be tested. That means the effort for testing remains even when the effort for designing and coding is eliminated. Of the two weeks for testing, 5 days will be for unit testing, i.e. testing the service in a simulated environment, and 5 days for integration testing, i.e. testing the service in the target environment together with the other existing services. Based on those assumptions the release delivery dates can be fixed at the start of a sprint and cannot be changed during the course of the sprint [Linz13].

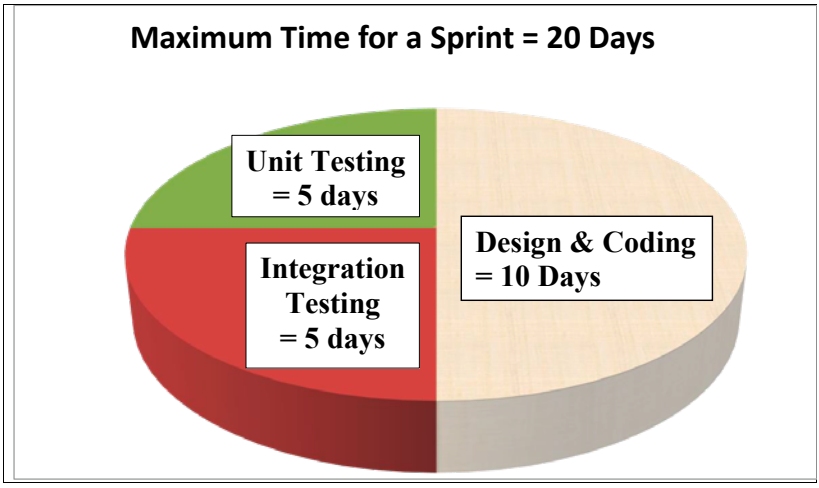


Figure 1: Micro service release cycle

This means the services to be delivered must be testable within these time limits. This is referred to as backward planning [Boeh03]. The size and complexity of the target service is determined by the effort and time required to test it. If the effort is too great and the time too long, the service size and complexity must be reduced according to the Boehm equation:

Effort = (Size * Complexity) / Productivity [Boeh99]

2 Conducting a Benchmark Test

If the services are newly developed ones, they should be designed to fulfill these criteria. If they are reengineered services, they must be refactored to fulfill the criteria. That means the designer or reengineer of the services must know how much code and how much functionality can be tested within a given time period. For this experiment 7 Java micro services with varying sizes were tested with 4 different coverage measurements

- Statement coverage
- Branch coverage
- Path coverage
- Parameter coverage [CrGr09].

The Java services used for the benchmark test were as follows:

1) a calendar conversion service ,
2) an order entry service ,
3) a savings-and-loan partner update service,
4) a beauty salon billing service,
5) a geometric form query service,
6) a user authorization service.
7) A bank mail service

The goal of the benchmark test was to measure the average effort required to execute a unit test case in a typical Java micro service. This effort in person hours was then multiplied by the complexity-adjusted number of test cases required to attain the desired test coverage of each target service. The results were as follows:

Service	Opers	Stmts	Logic Branches	Params	FuncPts	Test Paths	Tester Hours
Calendar	3	473	31	38	12	15	8
OrderEntry	16	625	187	43	29	92	37
BauSparer	17	276	47	64	35	22	13
BeautySalon	24	429	72	54	18	33	21
Geometry	5	510	73	19	9	36	18
Authorize	27	573	265	19	22	130	65
MailService	48	3317	762	211	126	278	88
Total	140	6203	1437	448	251	606	250

Table 1: Test costs

As is shown here the total test costs of all services were 250 hours with an average of 36 hours per service.

1) Calendar Service

For the calendar it took 8 hours to test the service with 3 operations, 473 statements, 31 branches, 38 parameters, 12 function-points and 15 test-paths, meaning that this service was well within the time limitation of 40 hours. Within the same time, we could have tested 4 services of a similar size and complexity.

2) Order Entry Service

For the OrderEntry it took 37 hours to test the service with 16 operations, 625 statements, 187 branches, 43 parameters, 29 function-points and 92 test-paths, meaning that this service was just within the time limitation of 40 hours. Within the same time, we could not have tested any other services of a similar size and complexity.

3) Bausparer Service

For the Bausparer it took 13 hours to test the service with 17 operations, 276 statements, 47 branches, 64 parameters, 35 function-points and 22 test-paths, meaning that this service was well within the time limitation of 40 hours. It would have been possible to test two additional services of a similar size and complexity.

4) BeautySalon Service

For the BeautySalon it took 21hours to test the service with 24 operations, 429 statements, 72 branches, 54 parameters, 18 function-points and 33 test-paths, meaning that this service was equally well within the time limitation of 40 hours, but it would not have been possible to test an additional service of a similar size and complexity without proscribing overtime.

5) Geometry Service

For the Geometry exercise it took 18 hours to test the service with 5 operations, 510 statements, 73 branches, 19 parameters, 9 function-points and 36 test-paths, meaning that this service was also well within the time limitation of 40 hours. It would have been possible to test an additional service of a similar size and complexity.

6) Authorization Service

For the Authorization it took 65 hours to test the service with 27 operations, 573 statements, 265 branches, 19 parameters, 22 function-points and 130 test-paths, meaning that this service was beyond the time limitation of 40 hours. It would be necessary to either lower the test coverage criteria by 60% or to split it up into two services to be able to test it in a single sprint. This is where agile testing requires compromises.

7) Mailing Service

For the Mailing it took 88 hours to test the service with 48 operations, 3317 statements, 762 branches, 211 parameters,126 function-points and 278 test-paths, meaning that this service required more than double the time limitation of 40 hours. Here it was out of the question to lower the test coverage. The service has to be cut up into two. This is where the time limitations of agile testing conflict with the current functionality.

3 Conclusions of the Benchmark Test

The quintessence of the study was that most micro services, at least those contained in this experiment, can be tested within the time limitations imposed by an agile scrum project. The time limitation of 40 hours can be transposed into circa 96 test cases, i.e. test paths, at the rate of 2.4 test cases per tester hour. The developers who cut the services out of existing code must see to it that the services are self- contained and do not require more than 96 test cases to be adequately tested. According to the analysis of the Java code for billing medical costs for

pensioners in Austria there were 2.745.224 lines of code, 1.094.684 statements, 262389 branches and 124.364 possible execution paths. By dividing the number of execution paths, i.e. procedural test cases, by the average test productivity taken from the bench mark study = 2.4 the total tester hours was calculated to be 51.818 tester hours or 2.590 tester days. Of course the test coverage criteria had to be lowered, so that in the end less than 1.940 tester days or 97 tester months were actually required. This only goes to show how costly testing can become, even in agile testing, if the test coverage goals are set too high. The service test of the health insurance services had cost 2.912 tester days before it was finally terminated with a branch coverage of 83%.

The most important conclusion is that micro services should not have more than 96 procedural test cases if they are to be tested within one week's time. That amounts to an average of 8.8 statements per test case or a maximum of 845 statements per service.

References

[Beck01] Beck, K. et al.: "Manifest for Agile Software Development", agilemanifesto.org/ iso.de, 2001

[Black99] Black, Rex: Managing the Testing Process, MicroSoft Press, Redmond, 1999

[Linz13] Linz, T. Testen in Scrum Projekten, dpunkt.verlag, Heidelberg, 2013

[Boeh03] Boehm, B./ Turner, R.: Balancing Agility and Discipline – A Guide for the Perplexed, Addison-Wesley, Reading, Ma., 2003

[Boeh99] Boehm, B. et al: Software Cost Estimation with COCOMO-II, Prentice-Hall, Upper Saddle River, New Jersey, 1999

[CrGr09] Crispin, L. / Gregory, J.: Agile Testing – A practical Guide for Testers and agile Teams, Addison-Wesley-Longman, Amsterdam, 2009

Call for Papers
Virtuelle Zusammenarbeit und verlorene Kulturen?

Trier, 8. & 9. September 2022

Achte gemeinsame Tagung der Fachgruppen Vorgehensmodelle und Projektmanagement im Fachgebiet der Wirtschaftsinformatik der Gesellschaft für Informatik e.V. (GI), in Kooperation mit der Fachgruppe IT-Projektmanagement der GPM e.V., sowie dem PMI Germany Chapter e.V.

Die Pandemie wirft gewohnte Muster durcheinander. Organisationen und ihre Mitarbeitende müssen sich für neue Arten der Zusammenarbeit öffnen. Die Diversität der Arbeitsweisen wird möglicherweise als eine der wichtigsten Erkenntnisse der Corona-Krise in die „bisherige" Arbeitswelt hineinreichen. Gefordert wird unter anderem, dass sich die Führungskultur vom Prinzip „Command-and-Control" hin zum Prinzip des gegenseitigen Vertrauens auf Basis professioneller Kooperationsarbeit wandeln muss. Das bedeutet besonders für Organisationen, die primär hierarchisch aufgebaut sind, eine gravierende Umstellung, speziell was die Führung der Mitarbeitenden betrifft.

Die Qualität der Kommunikation ist einer der wichtigsten Faktoren für die Zusammenarbeit im Team und den Erfolg von Projekten. Die Bindung der Teammitglieder untereinander über Ländergrenzen, Zeitzonen, ethnische und kulturelle Unterschiede hinweg zu erzielen und zu erhalten, ist die Königsdisziplin einer virtuellen Zusammenarbeit. Diese Bindung entsteht unter anderem durch gemeinsame Ziele, Häufigkeit und Qualität der Interaktionsbeziehungen und das Vertrauen der Teammitglieder untereinander.

Ungeklärt ist dabei, ob der Modus Operandi klassischer wie agiler Vorgehensmodelle in der digitalen Welt anders ist als in der analogen Welt. Obwohl die virtuellen Formate keine großen Unterschiede zu den Präsenzformaten haben, muss vieles neu gelernt und professionalisiert werden.

Was wird bleiben, wenn unser Leben nicht mehr von der aktuellen Pandemie beeinflusst wird? Wird es auch in der Arbeits- und Projektwelt zu „Long-Covid-Symptomen" kommen? Welche positiven Effekte wären wünschenswert?

Themenschwerpunkte

Im Mittelpunkt der diesjährigen Tagung PVM 2022 steht daher das Thema der virtuellen Zusammenarbeit und deren Auswirkungen auf kulturelle Themenstellungen. Uns beschäftigen die Fragen, inwiefern virtuelle Zusammenarbeit durch angepasstes Projektmanagement und angepasste Vorgehensmodelle unterstützt werden, und wie das Projektmanagement und Vorgehensmodelle ihrerseits von neuen Vorgehensweisen profitieren und dadurch weiterentwickelt werden können.

Um diese Fragen im Spannungsfeld zwischen Academia und Praxis zu diskutieren, laden wir in diesem Jahr insbesondere – aber nicht ausschließlich – zu Beiträgen zu folgenden Themenkomplexen ein:

- **Virtualisierung traditioneller Präsenzformate:** In diesem Themenbereich beschäftigt sich die Tagung mit der virtuellen Gestaltung von Workshop-Formaten, innovativen kollaborativen Formaten und neuen Formen für informale Rituale wie Ad-hoc-Gespräche. Haben sich die Beweggründe für Vor-Ort-Meetings hin zu Vehikeln der Beziehungspflege verschoben? Wie werden Motivation, Aufmerksamkeit und Mitwirkung in virtuellen Meetings sichergestellt? Besteht ein Bedarf für einen Knigge der digitalen Zusammenarbeit? Ist eine „neue Rhetorik" erforderlich, um im digitalen Raum zu überzeugen? Wie gestaltet sich der fachliche Austausch in virtuellen Teams und wie fließen Erfahrungen aus der (analogen) Arbeit ein? Wie kann osmotische Kommunikation im virtuellen Setting entstehen?
- **Kultur und Führung in virtuellen Settings:** In diesem Themenbereich beschäftigt sich die Tagung mit den Auswirkungen der Virtualisierung auf Facetten wie Kultur, Projektarbeit, Effektivität, Mitarbeiterbindung und Führung. Was macht Unternehmenskultur aus und wie ändert sich diese durch die virtuelle Zusammenarbeit? Was sind die Artefakte in agilen und hybriden Kooperationssettings? Wie teilen Wissensträger ihr Know-how innerhalb von Teams und darüber hinaus in der virtuellen Zusammenarbeit? Hat die Virtualisierung Auswirkungen auf die Diversität und Aspekte wie zum Beispiel Gendergerechtigkeit, Altersdiskriminierung und kulturelle Hintergründe? Sind in der virtuellen Zusammenarbeit andere oder neue Soft Skills notwendig? Müssen neue Wege im Stakeholder-Management gegangen werden? Erfordert ein möglicherweise eintretender „Digital Overload" auf der anderen Seite zeitweise ein „Digital Detox"? Wie organisieren sich Teams und wann ist welcher Führungsansatz zu präferieren? Existieren Situationen, in denen eine hierarchische Führung zu präferieren ist? Wie treffen Teams Entscheidungen, wie werden Konflikte gelöst und wie wird Verantwortung verteilt und geteilt?
- **Virtuelle Zusammenarbeit:** In diesem Themenbereich beschäftigt sich die Tagung mit den Auswirkungen, Möglichkeiten und Problemen der virtuellen Zusammenarbeit. Ermöglicht die Virtualisierung höhere Flexibilität in der Teamkonfiguration oder sind gerade hier langfristig eingespielte Teams von Vorteil? Wie verändert sich die Zusammenarbeit auch über Unternehmensgrenzen hinweg? Welche Möglichkeiten ergeben sich aus einer virtuellen Softwareentwicklung und einem "virtuellen Software Engineering"? Ermöglichen flexible, anpassbare virtuelle Welten eine effiziente und effektive Kollaboration oder sind das Spielereien, die viel Aufwand und wenig Nutzen verursachen? Welche psychologischen Faktoren und Risiken sind in der virtuellen Zusammenarbeit zu beachten? Wie sieht das zukünftige Metaversum für Projektmanagement und Softwareentwicklung aus? Welche Chancen, Risiken und Herausforderungen ergeben sich für Trainings und Schulungen durch Virtual Reality? Welche Anforderungen an den Datenschutz sind im Zusammenhang der AR/VR-Kollaboration zu beachten? Welche „Data Literacy" ist im Projektmanagement erforderlich? Wie definiert sich ein Digitaler Reifegrad im Projektmanagement und welche Methoden zu dessen Messung existieren? Wie beeinflusst die virtuelle Zusammenarbeit Vorgehensmodelle und Entwicklungsprozesse?

Ziele der Fachtagung

Ziel der Veranstaltung ist es, einem Fachpublikum fundierte Ansätze aus der Wissenschaft mit Erfahrungen zu deren Anwendung in der Praxis vorzustellen und Raum für die fachübergreifende Diskussion und den Erfahrungsaustausch zu geben.

Special Tracks

Future Track

Eine wichtige Aufgabe der GI-Fachgruppen ist es, sich mit der Zukunft des Fachgebiets zu beschäftigen. AutorInnen im Future Track können reifende Ideen oder kontroverse bzw. provokative Ansichten in einem Impulsbeitrag vorstellen, welche anschließend im Auditorium diskutiert werden sollen. Durch den Austausch sollten Denkanstöße und Impulse für die Teilnehmenden und auch die künftige Fachgruppenarbeit entstehen. Die Impulsbeiträge (nur als Kurzbeitrag, d.h. max. 5 Seiten) des Future Tracks werden einem separaten Review-Verfahren unterzogen.

Student Track

Ziel des Student Track ist es, gezielt Studierenden und NachwuchswissenschaftlerInnen (DoktorandInnen in einer frühen Phase) die Möglichkeit zu eröffnen, sich aktiv in die Tagung einzubringen und damit die Community der GI-Fachgruppen Vorgehensmodelle und IT-Projektmanagement kennenzulernen. Studierende und NachwuchswissenschaftlerInnen sind aufgerufen, qualitativ hochwertige Arbeiten einzureichen. Die Beiträge werden einem separaten Review-Verfahren unterzogen und als Kurz- oder Langbeitrag akzeptiert. Für akzeptierte Beiträge kann die Unterstützung bei der Finanzierung von Anreise, Tagungsgebühr und Unterkunft beantragt werden.

Journal First

Ähnlich dem etablierten „Journal-first"-Modell anderer internationaler Konferenzen, können auf der PVM 2022 Beiträge präsentiert werden, welche in renommierten Journalen und Konferenzen publiziert oder zur Publikation/Präsentation angenommen wurden. Ziel ist die Stimulation des Diskurses innerhalb der wissenschaftlichen Community sowie die Erhöhung des Impacts von bereits veröffentlichten Ergebnissen durch den Austausch mit Fachleuten aus der Praxis. Es werden ausschließlich Vorschläge von begutachteten Beiträgen akzeptiert, die auf der entsprechenden Hauptkonferenz (bzw. in Journalen) in voller Länge angenommen wurden.

Einreichung, Format und Fristen

Details zur Einreichung (Vorlagen, Seitenzahlbegrenzungen, Einreichungssystem) entnehmen Sie bitte unserer Website: https://pvm-tagung.de/Einreichung

Ein herausragender Beitrag wird mit dem Best Paper Award prämiert.

Es gelten folgende Fristen:

- *01.04.2022*: Einreichung eines vorläufigen Beitragstitels und ggf. Abstracts
- *15.04.2022*: Einreichung des Beitrags durch die Autoren
- *24.06.2022*: Benachrichtigung der Autoren
- *14.07.2022:* Einreichung finaler Version
- *01.09.2022:* Einreichung der Vortragsfolien

Tagungsband und Indizierung

Der Tagungsband wird in gedruckter Form in den GI Lecture Notes in Informatics publiziert (https://www.gi.de/service/publikationen/lni.html) und von dblp (http://dblp.uni-trier.de/) indiziert.

Tagungsort

Die Tagung findet am 8. und 9. September in Trier statt:

Universität Trier Campus II
Behringstraße 21
54296 Trier

Sollte die Pandemiesituation eine Präsenzveranstaltung nicht zulassen, ist die Durchführung als virtuelle Veranstaltung geplant.

Kontakt

Auf der Webseite http://pvm-tagung.de werden laufend aktualisierte Informationen zur Tagung bereitgestellt. Für Rückfragen wenden Sie sich bitte an info@pvm-tagung.de.

Für das Programmkomitee der Tagung:

Dr. Masud Fazal-Baqaie (Sprecher der Fachgruppe Vorgehensmodelle)
Dr. Enes Yigitbas (Stv. Sprecher der Fachgruppe Vorgehensmodelle)
Prof. Dr. Martin Engstler (Sprecher der Fachgruppe Projektmanagement)
Alexander Volland (Stv. Sprecher der Fachgruppe Projektmanagement)
Prof. Dr. Oliver Linssen (Sprecher der Fachgruppe IT-Projektmanagement der GPM)
Dr. Martin Bertram (Vorstandsmitglied PMI Germany Chapter)
Prof. Dr. Axel Kalenborn (Universität Trier)

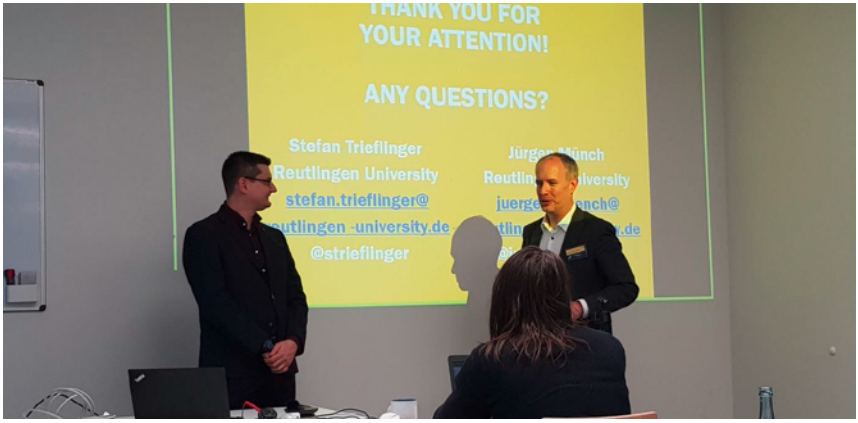
Tagungsbericht Software Management 2021
Software Management in Zeiten digitalisierter und vernetzter Produkte

von Andreas Helferich und Robert Henzel

Am 11. und 12. November fand die 13. Tagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung (WI-MAW) in den Räumlichkeiten der International School of Management (ISM) in Stuttgart statt. Unter dem Motto „Software Management in Zeiten digitalisierter und vernetzter Produkte“ fand die Tagung dabei erstmalig hybrid statt – mit einem Dutzend vor Ort Anwesenden sowie gut 20 per Webkonferenz zugeschalteten Teilnehmerinnen und Teilnehmern.

Nach kurzen Grußworten von Prof. Dr. Johannes Moskaliuk, dem Leiter des Campus Stuttgart der ISM sowie Prof. Dr. Andreas Helferich als Leiter des Organisationskomitees bildete die Keynote von Dr. Ulrike Dowie, Head of Analytics bei Siemens Data Visions, den Auftakt. Unter dem Titel „Künstliche Intelligenz (KI) in der Fabrik: Erfolgsfaktoren für KI-basierte Material- und Produktionsplanung“ berichtete sie von einem spannenden Anwendungsbeispiel künstlicher Intelligenz im produzierenden Umfeld und ging dabei besonders auf Erfolgsfaktoren und Management-Aspekte der Einführung und Wartung von KI-Lösungen ein.

Dimitri Petrik (Uni Stuttgart) stellte anschließend eine Multiple-Case-Study zur Nutzung von Social Media im Umfeld von Industrial Internet of Things (IIoT)-Plattformen vor, die den aktuellen Stand der Forschung im Bereich IIoT-Plattformen ergänzt. Stefan Trieflinger (Hochschule Reutlingen) stellte Ergebnisse aus seiner Forschung zu geeigneten Formaten für die Entwicklung und Handhabung von Produkt-Roadmaps in dynamischen und unsicheren Märkten.



Fragenrunde zum Vortrag von Stefan Trieflinger (li.) moderiert von Andreas Helferich (re.), Organisationskomiteeleitung

Der Vortrag von Dr. Claus Habiger (ITS Germany e.V.) zur Erweiterung eines bestehenden Standards für den Datenaustausch zur Unterstützung der Mobility as a Service Domänesowie daraus resultierende Schlüsse für die Rolle von Standards und Standardisierung für das Software Management schlossen den Vormittag des ersten Tags ab.

Der Nachmittag begann mit dem Vortrag von Dr. Christian Zinke (Uni Leipzig) zu „Digital Social Service Engineering“, der die Herausforderungen bei der Entwicklung und Integration von Softwarelösungen in die Prozesse der personenbezogenen Dienstleistungen in den Fokus stellte. Der anschließende Vortrag von Christopher Ringel (HS Heilbronn) hatte dann ein Best-Practice-Fallbeispiel für Software Engineering in der Mobilitätsbranche unter besonderer Berücksichtigung der Endnutzer als Experten über den gesamten Entwicklungsprozess hinweg und darüber hinaus zum Thema. Dr. Thomas Greb schloss den Engineering-Block mit seinem aus seiner umfangreichen Praxiserfahrung entwickelten systematischen Ansatz für das Tailoring hybrider IT-Projektmethoden ab.

Den Abschluss des Tages bildete die Sitzung des Fachausschusses WI-MAW, auf der Prof. Dr. Andreas Helferich und Dimitri Petrik zu den neuen Sprechern der Fachgruppe Software Produktmanagement gewählt wurden.

Der Freitag begann mit der Keynote von Prof. Dr. Michael Burmester (Hochschule der Medien Stuttgart), der von seiner Arbeit als Koordinator des Mittelstand 4.0- Kompetenzzentrum Usability berichtete und aufzeigte, wie das Erlebnis der Nutzung von Software möglichst positiv gestaltet und damit neue Innovationspotenziale geschaffen werden können. Prof. Dr. Ralf Kneuper (IU Internationale Hochschule) ging anschließend in seinem Vortrag auf aktuelle Entwicklungen im Datenschutz und ihre Bedeutung für das IT-Management ein. Prof. Dr. Franz Lehner (Uni Passau) reflektierte in seinem Vortrag die Software-Wartung über einen Zeitraum von vier Jahrzehnten und unternahm dabei den Versuch einer integrativen Bestandsaufnahme aus der Perspektive der Wirtschaftsinformatik und der Informatik. Den Abschluss des wissenschaftlichen Programms bildete der Vortrag von Robert Henzel (Uni Stuttgart) zu Erfolgsfaktoren für das IT-Produktmanagement in der Digitalen Transformation.

In Summe bot die Software Management 2021 trotz der durch die Pandemie bedingten Rahmenbedingungen einen guten Überblick über verschiedenste Aspekte des Software Managements, auch ergaben sich zu einigen Vorträgen sehr aktive Diskussionen unter Beteiligung von sowohl Präsenz- als auch Online-Teilnehmern und Teilnehmerinnen. Der persönliche Austausch fiel dennoch auch unter Berücksichtigung der Hygienemaßnahmen vor Ort einfacher, so dass nachvollziehbarer Weise der Wunsch besteht, die nächste Software Management im Jahr 2023 wieder in Präsenz durchzuführen.

Das Organisationsteam bedankt sich bei allen Beteiligten, insb. den Autoren, Reviewern sowie den Sponsoren auvesy GmbH und adkus e.V.! Aber auch der GI-Geschäftsstelle und dem Köllen-Verlag sei gedankt, ohne deren Unterstützung es nicht möglich gewesen wäre, den Tagungsband schon zur Konferenz vorliegen zu haben bzw. den Online-Teilnehmerinnen und -Teilnehmern zuschicken zu können!

Bewertung und Optimierung der Performance von Single Page Applications

Maximilian Bieleke, Andreas Schmietendorf

Hochschule für Wirtschaft und Recht Berlin
Email: andreas.schmietendorf@hwr-berlin.de

1 Motivation

Der Einsatz von Single Page Applications kann im Zusammenhang mit modernen Webapplikationen (z.B. bei Facebook) zunehmend beobachtet werden. Im Unterschied zu klassischen Webanwendungen, bei denen zwischen verlinkten Webseiten navigiert wird, erfolgt bei dieser Art von Anwendungen die algorithmische Verarbeitung der Präsentationsschicht mit Hilfe von nur einer HTML-Basisseite innerhalb des Webclients (z.B. Browsers, Web-Apps). Benötigte inhaltliche Veränderungen der Basisseite werden dynamisch vom Webserver über das Internet nachgeladen, ohne jedoch - wie im Falle einer verlinkten Webseite - einen neuen Sitzungszustand zu erzeugen.

Aus diesem Architekturprinzip resultieren vielfältige Vorteile, wie z.B. die Reduktion der Kommunikation zwischen Webclient und Webserver, die Entlastung des Webservers von HTML-Rendering- und UI-Logik, woraus eine verbesserte Skalierbarkeit resultiert, aber auch die Entkopplung der zu entwickelnden Client- und Serverkomponenten (d.h. zustandslose Kommunikation). Für die Anwender entsprechender Lösungen vergegenwärtigen sich die Vorteile in einer schnelleren Interaktionsfähigkeit, grafisch moderneren Nutzerschnittstellen oder auch in den einhergehenden Offline-Fähigkeiten der so entwickelten Anwendungen (insbesondere bei mobil eingesetzten Web-Apps). Nicht alle Anwendungsszenarien profitieren allerdings vom SPA-Architekturprinzip. Gerade im Diskurs klassischer Web-CMS-Lösungen, bei stark formularbasierten Anwendungen oder auch beim ggf. notwendigen Einsatz veralteter Browser-Systeme gilt es die Vor- und Nachteile einer SPA-Migration gegeneinander abzuwägen. Ein wichtiges Entscheidungskriterium sind dabei die erreichbaren Performanceeigenschaften konkreter Webanwendungen im Sinne der zeit- und ressourcenbezogenen Effizienz.

2 Themenbereiche

Eine an der HWR Berlin und bei der DB Systel angefertigte Forschungsarbeit setzte sich mit den Performanceaspekten von Single-Page-Applications (kurz SPA) aus Sicht der Softwareentwicklung, auseinander. Dabei wurde insbesondere auf die folgenden Themen eingegangen:

- Überblick zu den funktionalen und qualitativen Eigenschaften bzw. den Einsatzszenarien von Single-Page-Applications.
- Performance-Einflusskriterien auf Single-Page-Applications in der Entwicklung und beim Betrieb (u.a. Netzwerk, JS-Framework, Browser),
- Möglichkeiten zum Performance-Monitoring von Single-Page-Applications (Messgrößen und Messwerkzeuge - Latenzen, Memory-Leaks, loadtime, …),
- Umgang mit initialen und im Hintergrund auftretenden Ladezeiten (u.a. Kriterien für Client-Side vs. Server-Side-Rendering),
- Generischer Leitfaden zur Optimierung/Tuning des Performanceverhaltens von Single-Page-Applications (u.a. HTML, JS, CSS, Lazy-Loading),

- Quellcodebezogene Optimierungsansätze, wie z.B. Scriptsplatzierung (JS/CSS), DOM-Interaktionsverhalten, Einsatz von Ladeanimationen,
- Reflektion der gewonnenen Erfahrungen aus der Erprobung des Konzepts innerhalb eines Industrieprojektes.

3 Buchpublikation

Neben der Buchpublikation erfolgte die Präsentation der Forschungsergebnisse auch im Rahmen der ASQF net week 2022[1].

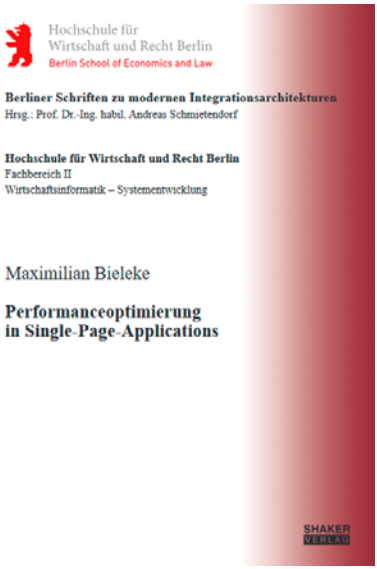


Abbildung 1: Buchpublikation [Bieleke 2021]

4 Quellenverzeichnis

[Bieleke 2021] Bieleke, M.: Performanceoptimierung in Single-Page-Applications, in Schmietendorf, A. (Hrsg.) Berliner Schriften zu modernen Integrationsarchitekturen, Shaker-Verlag, Düren, Dezember 2021, Band 26, ISBN 978-3-8440-8315-6

[BielSchmie 2022] Bieleke, M.; Schmietendorf, A.: Performanceaspekte von Single-Page-Applications aus Sicht der Softwareentwicklung, fortgeschrittenen Vortrag im Rahmen der 5. ASQF Net Week, 25. März 2022

[1] https://www.asqf.de/eventkalender/asqf-net-week/

Der Fachausschuss und die Fachgruppen WI-VM, WI-PM, WI-PrdM stellen sich vor

Fachausschuss WI-MAW:
Management der Anwendungsentwicklung und -wartung

Anwendungssysteme sind aus Sicht der Wirtschaftsinformatik Aufgabenträger im Rahmen der Erfüllung der betrieblichen Gesamtaufgabe. Ihre Aufgabenstellungen werden aus den Unternehmenszielen und den strategischen Zielen der Informationsverarbeitung abgeleitet. Die Entwicklung von Anwendungssystemen erfolgt nicht "kontextfrei", sondern i.A. in einem bestimmten betrieblichen Umfeld. Dies bedeutet zum einen, dass sich das einzelne Anwendungssystem in bereichsübergreifende bzw. unternehmensweite Daten- und Funktionsmodelle oder Objektmodelle einordnen muss. Zum anderen existieren häufig bereits Anwendungen für andere betriebliche (Teil-)Aufgaben, mit denen das System zusammenarbeiten muss.

Der Fachausschuss beschäftigt sich aus dieser Sicht mit der Planung, der Entwicklung, der Einführung, dem Einsatz und der Wartung betrieblicher Anwendungssysteme. Im Vordergrund stehen Vorgehensweisen, Prinzipien und Methoden für die Anwendungsentwicklung im betrieblichen Umfeld sowie ihre Unterstützung durch Softwarewerkzeuge. Im Einzelnen setzt sich der Fachausschuss mit Themen wie den folgenden auseinander:

- Integration von Anwendungssystemen in eine existierende betriebliche DV-Landschaft;
- Sicherung der Investitionen in das Wirtschaftsgut Software; Bewertung von Vorgehensmodellen, Methoden und Werkzeugen zur Anwendungsentwicklung sowie Einsatzerfahrungen;
- Management von Softwareentwicklungsprojekten (Projektplanung, -durchführung und -kontrolle, Projektorganisation, Projektmanagementsysteme, Kosten/ Wirtschaftlichkeit),
- Software Produktmanagement, Configuration Management, Change Management, Migration Management, Reengineering.

Mitgliederzahl: ca. 500

FA-Sprecher

Prof. Dr. G. Herzwurm
Universität Stuttgart
Lehrstuhl für Allgemeine
Betriebswirtschaftslehre und
Wirtschaftsinformatik II
(Unternehmenssoftware)

stellv. FA-Sprecherin

Dr.-Ing. Birgit Demuth
Technische Universität Dresden
Institut für Software- und
Multimediatechnik

Fachgruppe WI-VM:
Vorgehensmodelle für die betriebliche Anwendungsentwicklung

Betrachtungsgegenstand der Fachgruppe sind die als "Vorgehensmodelle" bezeichneten Beschreibungen der Aufbau- und Ablauforganisation von Projekten zur Entwicklung und Wartung von Anwendungssystemen. Solche Beschreibungen helfen, die Durchführung von Projekten innerhalb eines Unternehmens oder darüber hinaus zu standardisieren und zu verbessern. Der Begriff Anwendungssystem sei hier sehr weit gefasst: von technischen über betriebswirtschaftliche bis zu organisatorischen Systemen.
Um eine effektive und effiziente Gestaltung der Vorgehensmodelle und damit der Projekte zu erreichen, ist die Berücksichtigung der Schnittstellen zur Betriebswirtschaftslehre einerseits, insbesondere der Organisations- und der Managementlehre, und dem Software Engineering andererseits wesentlich.
Das Thema "Vorgehensmodelle" wird daher von der Fachgruppe aus verschiedenen Blickrichtungen betrachtet:

- Grundlagen: Begriffsdefinitionen, Bestandteile, (formale) Beschreibung von Vorgehensmodellen, Vorgehensmodell-Typen.
- Inhaltliche Bausteine: Konzepte, Methoden, Phasen, Projektmanagement, Qualitätssicherung.
- Werkzeugunterstützung: Vorgehensmodell-Driver, Meta-Modelle, Data-Dictionaries.
- Ökonomische, soziale und psychologische Aspekte: Einführung und Betrieb von Vorgehensmodellen, organisatorisches Umfeld.
- Beispiele aus der Praxis: Standard-Vorgehensmodelle in Organisationen, Branchen und für Anwendungstypen, spezielle Vorgehensmodelle von Unternehmen.
- Standardisierung von Vorgehensmodellen: V-Modell XT, Hermes

Die Fachgruppe fördert einen intensiven Gedankenaustausch durch die Pflege persönlicher Kontakte und unterstützt einen offenen und kritischen Dialog zwischen Wissenschaft und Praxis. Ein weiteres Ziel der Fachgruppe ist die Erarbeitung von Empfehlungen und Stellungnahmen zu den technischen, wirtschaftlichen, organisatorischen und sozialen Aspekten bei Auswahl und Einsatz von Vorgehensmodellen - dies insbesondere vor dem Hintergrund nationaler, europäischer und internationaler Normungs- und Standardisierungs-bestrebungen.
Weitere Informationen über Vorgehensmodelle und die Arbeit der Fachgruppe sind im Internet zu finden unter www.vorgehensmodelle.de.

FG-Sprecher

Dr. rer. nat. Masud Fazal-Baqaie
Next Data Service AG
Berlin

stellv. FG-Sprecher

Dr. rer. nat. Enes Yigitbas
Fachgruppe Datenbanken- und Informationssysteme
Universität Paderborn

Fachgruppe WI-PM:
Projektmanagement

Die Fachgruppe befasst sich mit dem Einsatz, der Verbreitung sowie der Weiterentwicklung des Projektmanagements. Neben Vertretern aus den Hochschulen sollen vor allem Praktiker die Arbeitsschwerpunkte der Fachgruppe definieren, Ergebnisse erarbeiten und Erfahrungen weitergeben. Für die Aufgabengebiete des Projektmanagements sollen Methoden, Werkzeuge und Techniken untersucht werden. Neben den klassischen Aufgabengebieten wie beispielsweise Projektorganisation, Aufwandschätzung, Projektverfolgung und Projektsteuerung stehen folgende Themen im Vordergrund:

Bedeutung und Dimensionierung des Projektmanagements.
Die Bedeutung des DV-Projektmanagements als entscheidender Faktor für den Erfolg oder das Mißlingen von DV-Projekten wird von vielen Entscheidungsträgern unterschätzt. Daher sollte die grundsätzliche Bedeutung sowie der Nutzen einer angemessenen Ausstattung des Projektmanagements mit eigenen Ressourcen transparent gemacht werden.

Human Factors.
In zahlreichen Projekten liegen die größten Projektrisiken bei den sogenannten Human Factors (oder "weichen" Faktoren). Der Umgang mit solchen Risiken erfordert Kompetenz bei Themen wie Motivation, Führung, Teamfähigkeit, Überwindung "politischer" Widerstände u.a.m.

Programm Management.
Immer öfter gefordert wird das Management eines Portfolios von Projekten, wobei nicht alle Projekte des Portfolios eigentliche DV-Projekte zu sein brauchen. Solche Projektportfolios können beispielweise als Folge einer veränderten Unternehmensstrategie entstehen und sollen dann einen größeren Veränderungsprozess bewirken. Hauptaufgabe eines Programme Managements ist dabei die zielorientierte Steuerung der Abarbeitung des Projektportfolios, wobei insbesondere unternehmerische Gesichtspunkte zu beachten sind.

FG-Sprecher

Prof. Dr. Martin Engstler
Hochschule der Medien Stuttgart

stellv. FG-Sprecher

Alexander Volland
Union IT-Services GmbH
Frankfurt am Main

Fachgruppe WI-PrdM:
Software Produktmanagement

Effizientes und effektives Management softwareintensiver Produkte ist zu einer kritischen Kernkompetenz von Unternehmen geworden. Unternehmen sind mit einer stetig wachsenden Anzahl von Herausforderungen konfrontiert, die durch unterschiedliche Lebenszyklen von Systemen und unterschiedliche Kritikalität im Systemeinsatz in immer mehr – und neuen – Anwendungsfeldern entstehen. Hybride Systeme, z.B. im Internet-of-Things, in Automobilen, Flugzeugen, Drohnen, medizinischen Geräten oder in der Unterhaltungselektronik geben Software eine nie dagewesene Bedeutung. Zusätzlich entstehen durch die vielfältigen Initiativen im Rahmen Digitalisierung neue Arbeits- und Geschäftsmodelle und eröffnen vollkommen neue, durch Software getriebene Möglichkeiten zur Innovation.

In diesem dynamischen Umfeld findet softwaregetriebene Innovation an der Schnittstelle zwischen Informatik/Software Engineering und Wirtschaft statt, zwischen Forschung und industrieller Praxis. Das Produktmanagement umfasst hierbei die Entwicklung, Wartung und Evolution klassischer Softwarelösungen im gesamten Produktlebenszyklus, aber insbesondere auch innovative softwarebasierte Innovation. Die Fachgruppe befasst sich einerseits mit Konzepten, Methoden und Werkzeugen der Informatik/Wirtschaftsinformatik zur Gestaltung des Produktmanagements und der Produktinnovation. Andererseits wird insbesondere auch ein starker Fokus auf die praktische Anwendbarkeit theoretischer Konzepte gelegt.

Die Fachgruppe fördert auf dem genannten Gebiet den intensiven Gedankenaustausch, die Pflege persönlicher Kontakte und die Zusammenarbeit interessierter Personen und Gruppen. Dazu zählt u.a. die gegenseitige Information über Veranstaltungen, Projekte und Veröffentlichungen.

FG-Sprecher

Prof. Dr. rer. pol. Andreas Helferich
ISM International School of Management
Campus Stuttgart

stellv. FG-Sprecher

Dr. Dimitri Petrik
Universität Stuttgart
ABWL und Wirtschaftsinformatik II
(softwareintensive Business)

Mitglieder des Fachausschusses Management der Anwendungsentwicklung- und wartung (GI-MAW)

Die Mitglieder des Leitungsgremiums des Fachausschusses finden Sie unter:

https://fa-wi-maw.gi.de/fachausschuss/leitungsgremium

Impressum

Der Rundbrief des Fachausschusses *Management der Anwendungsentwicklung und -wartung (WI-MAW)* ist das Publikationsorgan des Fachausschusses sowie der Fachgruppen

WI-VM *Vorgehensmodelle für die betriebliche Anwendungsentwicklung*
WI-PM *Projektmanagement*
WI-PrdM *Software Produktmanagement*

Der Rundbrief erscheint einmal jährlich elektronisch. Durch den Rundbrief sollen wichtige Erfahrungen, neue Erkenntnisse und aktuelle Informationen unter den Mitgliedern ausgetauscht werden. Rundbriefbeiträge von Mitgliedern und Interessenten sind daher besonders willkommen. Es können Beiträge zu folgenden Rubriken eingereicht werden:

- Fachbeiträge: *Erfahrungsberichte; Theoretische Beiträge; Projektberichte (auch über laufende Projekte)*
- Informationen: *Buchbesprechungen; Tagungsberichte; Vorstellung von Arbeitsgruppen;*
- Leserbriefe: *Veranstaltungen; Call for Papers; Einladungen; Programme*

Es wird gebeten, Beiträge in elektronischer Form (Word) an die Rundbriefredaktion zu senden. Ein Ausdruck sollte keine Seitennummerierung enthalten, wegen der Verkleinerung auf DIN A5 jedoch eine Schrift von mindestens der Größe wie Times Roman 12.

Die Beiträge können in deutscher oder englischer Sprache abgefasst sein. Mit der Zusendung eines Beitrags ist das Einverständnis zur Veröffentlichung im Rundbrief verbunden. Jeder Beitrag wird ohne Begutachtung veröffentlicht.

Herausgeber Fachausschuss *Management der Anwendungsentwicklung und -wartung*
Auflage 500
Redaktion Christian Kop
Institut für Artificial Intelligence and Cyber Security
Universität Klagenfurt
A-9020 Klagenfurt
E-mail: christian.kop@aau.at
Tel.: +43 463 2700 3735
Fax: +43 463 2700 993735

Redaktionsschluß für das nächste Heft: 31.01.2023