

# Lebenslauf

Name: **Thomas Menzel**  
Adresse: Hasenwinkel 22  
31275 Lehrte OT Hämelerwald  
  
Mobil: 0152 293 682 74  
Email: [tmenzel@tomsit.de](mailto:tmenzel@tomsit.de)  
Web: [www.tomsit.de](http://www.tomsit.de)  
Geburtsdatum: 3.6.1974  
Geburtsort: Weißenhorn, Bayern  
Zivilstand: verheiratet  
Anzahl Kinder: 4  
Sprachen Deutsch (Muttersprache) und Englisch (wie Muttersprache)

## ***Kernkompetenzen:***

- Backendentwicklung von (Micro)services in Java mit Spring Boot
- Automatisierte (Integrations-)Tests
- CI/CD
- Cloud-Architektur

## ***Berufliche Stationen***

**07/2012 – heute**     **Software-Engineer, -Architekt, Teamlead, Techn. Projektleiter**  
Freelancer

**04/2005 – 07/2012**     **Software-Engineer, -Architekt, Projektleiter**  
Angestellt bei BROX IT Solutions GmbH, Hannover

**04/2000 – 04/2005**     **Software-Engineer** (Teilzeitanstellung neben Studium)  
BROX IT Solutions GmbH, Hannover

**04/2000 – 03/2006**     **Studium der Elektrotechnik/Technische Informatik**  
(Fortsetzung)  
Studienarbeit: Entwicklung von Algorithmen zur Fehleremulation nichtklassischer Schaltungsfehler in C++  
Fachpraktikum: Linux basierte Embedded Systems für DSP-Anwendungen bei Thomson in Hannover  
Diplomarbeit: Implementierung und Test von Multiplizierer- und Shifter-Modulen im Vektor-Pfad eines DSP in VHDL.

**04/1999 – 04/2000**     **Software-Engineer** (Vollzeitangestellter)  
BROX IT Solutions GmbH, Hannover

**10/1996 – 04/1999**     **Studium der Elektrotechnik/Technische Informatik**  
Universität Hannover

## ***Anhänge***

- [Projekte](#)
- [Skill Matrix](#)

## Projekte

10/2022 – heute

### GVL mbH – Verarbeitung von File- als auch Event-basierten Datenströmen

Rolle: Senior-Entwickler

#### Technologien

- Spring Boot Microservice (noch 2.7, angehend 3.x)
  - Teilweise wird der Reactive Stack verwendet
  - Java 11 (aktuell Migration auf 17)
  - Lombok
- API First Ansatz mit OpenAPI 3 + Code Generatoren
- RabbitMQ
- MongoDB 4.4, Posgre SQL
- Junit5, Mockito, Spring Test Framework
- Docker, Kubernetes, AWS
- Gitlab, Graylog, Grafana, New Relic, Prometheus
- Jira, Confluence

Bei der GVL habe ich ein sehr professionell aufgesetztes Entwicklungsteam kennengelernt, in dem die Rollen Product Owner (Anforderungsmanagement), Entwicklung, Testing und Operations durch dedizierte und sehr kompetente Kollegen besetzt sind. Damit konnte ich mich weitestgehend auf die Entwicklung der Anforderungen konzentrieren, was sehr zur Entwicklungsgeschwindigkeit beigetragen hat.

Im Rahmen der Entwicklung konnte ich denn auch einigen Wissensaufbau verzeichnen:

- Pro/Con Reactive Stack
- Mongo DB
  - Pro/Con Dokumentenbasierte Speicherung vs. Relationale DB
  - Queries und ein bisschen Aggregations
- Eventdriven Architecture mit AMQP (RabbitMQ)
- Grenzen beim Einsatz von OpenAPI 3
- Kleinteilige Stories und Fokus bei der Entwicklung auf diese
- Echter Microservice-Ansatz

04/2021 – 09/2021

### TUI: Design und Entwicklung eines Microservices für Reaktive Datenverarbeitung in einem Hochlast-Szenario

Rollen: Solution-Architekt, Lead-Developer

#### Technologien

- Kafka
- AWS Kinesis + KCL
- DynamoDB
- Spring Boot Microservice
- Junit5, Cucumber
- Java 11

- RactiveX 3
- Terraform
- Kubernetes + Flux

Ich bin in das Projekt ein paar Monate nach dessen Start gekommen. Erste maßgebliche Aufgabe war es, den per Trial-And-Error gewachsenen Service in strukturierte Funktionseinheiten und Abläufe zu refaktorisieren. Wesentliche Teile dessen waren:

- Konzeption
  - Klärung der Lücken in den Anforderungen und des Verhaltens der Randsysteme (diese waren bis Juli 2021 noch nicht implementiert)
  - Anpassung und Erweiterung des bestehenden Lösungskonzeptes, um die neuen Erkenntnisse zu berücksichtigen
- Aufräumen der ReactiveX Implementierung inkl. Migration nach V3
- Umstellung von einer einfachen AWS SDK Kinesis-Consumer Implementierung auf AWS KCL, um u.A. so automatisch auf Änderung der Shards reagieren zu können.
- Stabilisierung und Ordnung der Test sowie Aufbau der CI/CD Pipeline
- Aufbau eines Testkonzepts und Implementierung von Integrations-Tests mit Cucumber sowohl für den eigenen Service als auch den nachgelagerten, so dass beide als Einheit getestet werden können.

**12/2020 – heute  
(Nebenprojekt)**

### **Implementierung einer Applikation zur Produkt-Konfiguration und Verwaltung in einem Nischenmarkt**

Rollen: Solution-Architekt, Lead-Developer, Analyst

#### Technologien

- Spring Boot Backend-Service mit MariaDB
- Frontend mit Eclipse Scout (22.0.41)
- Gitlab

Hauptbestandteil meiner Arbeit war den existierenden Entwurf des Datenmodells, der sich stark an den bisherigen Excel Tabellen orientierte, zu „überarbeiten“. Dieses mündete schnell in das Einvernehmen mit dem Kunden, ein ganz neues zu schaffen und erforderte somit die Einarbeitung in die Anforderungen und Bedürfnisse der Zielanwendung. Im Rahmen der Modellierung und Implementierung des neuen Modells war eine beständige Überprüfung der technischen Machbarkeit im Zusammenhang mit den gewählten Technologien von Nöten (JSON via REST API definiert über OpenAPI sowie Code Generierung des Clients mit Jersey2 auf der einen Seite und JPA/Hibernate auf der DB Seite). Seither bin ich Ansprechpartner für die Umsetzung und Nutzung des Modells für die weitere Entwicklung.

Das Modell sollte u.A. dabei folgende Aspekte lösen:

- Abbildung von maschinenabhängigen Parametern auf bestimmte Attribute eines Produktes
- Mandatenfähigkeit in einer Unternehmensgruppe
- Mehrsprachigkeit auf DB-Ebene

Daneben habe ich maßgeblich für die folgenden Themen den Lösungsansatz konzeptioniert und in Teilen implementiert:

- Migration der Bestandsdaten in das neue Modell
- Autom. Berechnung von Produkt-Kennwerten als auch deren Toleranzen auf Basis von Konfigurations-Daten, die pro Produkt-Attribut wahlweise übersteuert werden können.

Im Laufe der Zeit ist auch ein wenig Entwicklung im Frontend mit dem Eclipse Scout UI Komponenten-Framework dazugekommen.

**04/2018 – März 2021**

### **Customer Master Data Management für TUI auf Basis von Spring Boot Microservices in AWS**

Rollen: Co-Solution-Architekt, Lead-Developer, DevOps Engineer

#### Tätigkeiten

- Design und Entwicklung der fachlichen Module für Data-Cleansing/Validation sowie Suche und Matching der Kunden
- Design und Aufbau der Macro-Architektur, d.h. der AWS Infrastruktur sowohl für die Applikation an sich als auch Build-Tooling (CI/CD) mit Blick auf Wartbarkeit und Applikation-Support (Logging, Observability) via Terraform

- Technische Konzeptionierung, Entwicklung, Unit-, Integrations- und End-2-End-Tests
- Jira Task Management, d.h. Strukturierung der Aufgaben in Epics, Stories, Tasks unter Nutzung von Links zur Abbildung der Abhängigkeiten und Kategorien
- Teamgröße: 2-6
- Agile Entwicklung ("TUI Scrum")

#### Technologien

- 9 RESTful Microservices auf Basis von Spring Boot 2.3, Maven, Java 11
- Hibernate, JUnit 5, Mockframeworks, PostgreSQL 10, JAXB, Swagger, Elasticsearch, Angular JS; Spring Sleuth
- AWS Cloud Komponenten mit Setup via Terraform: ECS, ALB, Security Groups, IAM, Cognito, Elasticsearch, Kibana, RDS/Aurora PostgreSQL, lambda, Parameter Store, VPC, S3, CloudWatch, X-Ray, Beanstalk, EC2
- Container: Docker
- CI/CD: Gitlab, davor: CodeCommit, CodeBuild, CodePipeline
- TDD, CleanCode

Fachlich geht es bei dieser neue zu schreibenden Applikation um eine eventbasierte Deduplizierung und Zusammenführung von Kunden aus verschiedenen Systemen. Damit wird *eine Quelle* für die Stammdaten aller Kunden für alle anderen Systeme geschaffen, die perspektivisch die bestehende Haupt-Kunden-DB ablösen soll.

Zu Beginn des Projektes bestand meine Hauptaufgabe im Aufbau der AWS Infrastruktur und der CI/CD Pipeline. Das war für mich (und auch TUI) komplettes Neuland und insofern war ich hier auf mich alleingestellt dieses aus der Taufe zu heben. Nach einem Jahr bekam ich endlich Unterstützung seitens der TUI durch einen Spezialisten in diesem Umfeld, so dass wir dann einige Verbesserungen vornehmen konnten. Daraus ergab sich, dass ich auch hauptzuständig für die Deployments als auch den Betrieb bin.

Neben dieser Aufgabe habe ich maßgeblich für den „Integrationskitt“ der Anwendung gesorgt, d.h. Konfiguration der Spring Boot Anwendung für alle Stages, d.h. fachliche als auch technische Konfigurationen (Spring Context/Beans) inkl. den Integrationen in die AWS Services sowie mit MDM verbundenen Systemen der TUI.

Neben diesen eher rein technisch getriebenen Arbeiten habe ich die fachlichen Module der Kunden-Suche und des Regelbasiertes-Matching konzeptioniert und entwickelt, später dann auch das Modul zum regelbasierten Cleansing der Daten übernommen.

Auf Teamebene habe ich es zu meiner Aufgabe gemacht (es fehlten sowohl ein Scrum-Master als auch ein Product Owner), die Entwicklungs-Arbeiten in sinnvolle Schritte zu strukturieren und als Jira-Tickets zu organisieren, mit dem Ziel das Vergangene nachvollziehbar zu halten, die Zukunft planbar zu

**03/2017 – 03/2018**

machen und Abhängigkeiten festzuhalten.

**J2EE / RCP Applikation „BIN“ für Talanx/HDI**

**Tätigkeiten**

- Technische Konzeptionierung, Entwicklung, Unit- und Integration-Tests im Bereich Dokumente und Drucksteuerung
- Teamgröße: ~12
- Projektgröße: >50 Entwickler
- Agile Entwicklung

**Technologien**

- Websphere, Java 6, Oracle 11g, JPA/Hibernate 4, Eclipse RCP, SOAP Webservices over JMS, EJB 2, Maven, SonarQube

BIN wird als spartenübergreifendes vertragsbestandsführende System für die Talanx-Mitarbeiter weltweit aufgebaut. Durch die schiere Größe des Projektes (Umfang der Funktionen, Projektmitarbeiter) hat es eine hohe Komplexität und nicht zuletzt, weil es notwendigerweise eine Vielzahl von Schnittstellen in die Umsysteme unterhält. Meine Aufgabe im Modul „Dokumente und Drucksteuerung“ lag vornehmlich in der Analyse der Anforderungen (meist 2-Zeiler aus dem Fachbereich), einem technischen Lösungskonzept sowie dessen Implementierung. Dies erforderte auch immer software-archäologische Untersuchungen der aktuellen Funktionsweisen. Darüber hinaus habe ich eine Vielzahl von Anregungen eingebracht, um die Entwicklungsarbeit zu verbessern (Dokumentation, Schulungen, Tooling, Coding Standards, ...).

**07/2016 – 03/2017**

**Webapplikation „Destimint“ für TUI Deutschland**

**Tätigkeiten**

- Technische Konzeptionierung, Entwicklung, Unit- und Integration-Tests
- Teamgröße: 9
- Agile Entwicklung

**Technologien**

- Java 8, Tomcat 8, Spring 4, Vaadin 7, Oracle 11g, JPA/Hibernate 5

Destimint dient TUI als zentrale Anwendung für interne Verwaltungsprozesse sowohl der eigenen Hotels als auch aus externen Quellen. Hierbei ist es in die Systemlandschaft integriert und unterhält bidirektionale Schnittstellen in die Buchungssysteme der TUI.

**08/2012 – 12/2016**

**Webapplikation „Engasstool“ für die VW Inbound-Logistik**

**Tätigkeiten**

- Technischer Projektleiter/Team-Lead
- Konzeption, Entwicklung, Release-Management
- Teamgröße: 4-6
- Scrum seit Jan 2016

**Technologien**

- Spring 4.x, Tomcat 7, WebSphere 8, JSF 2.1, PrimeFaces 5, Solr 3.6, Oracle 11g, STPs, JMS, JPA/Hibernate 4.3

Übernahme der technischen Team- und Projektleitung für eine Webapplikation mit Ziel dem Fachbereich ein Werkzeug für die Engpasssteuerung an die Hand zu geben. Zusätzlich zur Leitung der Entwicklung (auch als Software-Architekt und Anleitung der Teammitglieder via Code-Review) bin ich maßgeblich in die konzeptionelle Weiterentwicklung der Applikation in Absprache mit dem Fachbereich eingebunden (Rolle Product Owner, Analyst). Seit 2015 Q2 habe ich wieder vermehrt entwickelt, sowohl im Front- als auch Backend. Ab 2016/07 nur noch beratend einige Stunden im Monat tätig.

**07/2012 – 01/2013**

**Suche basierend auf eccenca**

**Tätigkeiten**

- Konzeption und Architektur der Weiterentwicklung
- Implementierung
- Teamgröße: 1-2

**Technologien**

- eccenca/SMILA
- Solr 3.5

Weitere Generalisierung des AG-eigenen Frameworks zur gleichzeitigen Einbindung mehrerer Suchtechnologien.

**03/2012 – 06/2012**

**Suche basierend auf eccenca für ein internes Web-Portal bei VW**

**Tätigkeiten**

- Konzeption und Architektur der Weiterentwicklung
- Implementierung
- Teamgröße: 1-2

**Technologien**

- eccenca/SMILA
- Google Search Appliance 6.14

Umsetzung neuer Features im Framework des AG und Einbindung der GSA als Suchtechnologie.

11/2011 – 06/2012

**Liferay CMS Suche basierend auf eccenca für ein internes Web-Portal bei VW**

Tätigkeiten

- Konzeption und Architektur
- Implementierung
- Teamgröße: 2-3

Technologien

- Liferay 6
- Solr 3.5
- eccenca/SMILA

Für den Such-Teil einer Search-Driven Liferay Applikation soll eccenca zum Zuge kommen. Hierbei gilt es ca. 80.000 gleichzeitige Nutzer bedienen zu können, was in der Spitze zu geforderten 1.1 Mio. (technischen) Such-Anfragen pro Stunde führt bei einem Dokumentenvolumen von anfänglich ca. 100.000.

Meine Aufgabe hierbei war zusammen mit dem Projektleiter zunächst die Anforderungen des Kunden und deren Umsetzung im Pflichtenheft festzuhalten. Umsetzung der Spezifikation, Design der SW-Architektur sowie deren Leitung und Qualitätssicherung oblagen mir.

03/2011 – 10/2011

**Suche basierend auf eccenca für das Intranet-Portal bei VW**

Tätigkeiten

- Konzeption und Architektur der Weiterentwicklung
- Implementierung
- Teamgröße: 1-2

Technologien

- eccenca/SMILA
- Lucene, Solr 1.4

Das Projekt war als Weiterentwicklung angelegt und basierte zu dem Zeitpunkt auf einer frühen eccenca/SMILA Version, die stark für die Bedürfnisse des Projektes modifiziert wurde. Ziel war es zum einen den Code auf eine aktuelle, unangepasste SMILA Version zu migrieren sowie neue Quellen zu integrieren und für die Suche zugänglich zu machen. Meine Aufgabe bestand im Wesentlichen darin, den bestehenden Code zu analysieren und fortzuschreiben, ohne die Möglichkeit auf das Wissen der früheren Programmierer zugreifen zu können. Trotz dieses widrigen Umstandes gelang es durch effektive Zusammenarbeit mit dem AG die neuen Anforderungen zur vollen Zufriedenheit abzubilden und innerhalb der von mir geschätzten Aufwände zu verbleiben. In dieser fruchtbaren Zusammenarbeit wurde das Framework des AG weiter generalisiert, um zukünftig einfach Quellen anbinden zu können.



**02/2011 – 07/2012**

**Eclipse Projekt SMILA und BROX eccenca**

Tätigkeiten

- SMILA Co-Projekt-Leiter
- eccenca Projekt-Leiter und Architekt
- Implementierung
- Teamgröße: 2-3

Technologien

- Integration Suchserver Solr 1.4
- eccenca 2.x
- SMILA

eccenca 1.0 war noch eine Hybridlösung bestehend aus dem neuen SMILA Indexierungs-Processing und den AnyFinder-Bestandteilen der Suche basierend auf Lucene. Mit eccenca 2.0 wurde dieser ältere Code ebenfalls durch SMILA ersetzt und zugleich Lucene durch Solr 1.4. Für die Architektur, Umsetzung und Weiterentwicklung von eccenca (und SMILA zum Teil) zeichne ich nun auf technischer Seite verantwortlich.

**07/2010 – 01/2011**

**Eclipse Projekt SMILA und BROX eccenca**

**Unified Information Access Architecture**

Fortführung der Arbeiten wie zu Beginn des Jahres

03/2010 – 07/2010

### **Commundo Tagungshotels: Tagungs-Konfigurator Großkunden**

Regelbasierte Web Applikation für Großkunden zur Buchung von Schulung inkl. Integration in Hotel ERP Software

#### Tätigkeiten

- Konzeption und Architektur
- Techn. Projektleiter Entwicklung, später Gesamt-Projektleiter
- Implementierung
- Teamgröße: 8 (2-3 Intern 5 Offshore)

#### Technologien

- Tomcat, SQL Server
- Spring 3
- JPA via Hibernate 3.3
- Sihot (Hotel ERP)
- Ontobroker 6.1

Das Projekt habe ich über seine ganze Laufzeit begleitet. An deren Beginn stand die Konzeption in Form von Lasten- und Pflichtenheft, die ich mit meinem Kollegen gemeinschaftlich verfasst habe. Hierbei habe ich mich auf die Prozesse und Architektur (Datenbank-Design, externe Schnittstellen) fokussiert, während mein Kollege für die UI zuständig war. Ebenso haben wir uns auch die Projektleitung geteilt, in der ich für die technischen Aspekte verantwortlich zeichnete. Dies beinhaltete vornehmlich die Kommunikation mit den Ansprechpartnern der zu integrierenden Systeme sowie mit dem Offshore-Entwicklungspartner (UI Programmierung) bzgl. der geforderten Funktionsweise und dem Zusammenspiel mit den übrigen Komponenten. In diesem Zusammenhang hat es sich als sehr vorteilhaft erwiesen, dass ich das DB-Design/JPA Objekt-Modell vollständig vorgegeben und implementiert habe und ebenso die Implementierung der Integration in das Sihot ERP System bzw. den Ontobroker. Im Rahmen dieses Projektes habe sich die folgenden Dinge gezeigt:

- Eine gute Spezifikation ist die halbe Miete
- Eine Offshore-Entwicklung geht gut, wenn eine hohe Verzahnung der Entwicklungsarbeit und -Teams gegeben ist, so dass die Kommunikationswege kurz sind.
- Entwicklung mit JPA und Spring
- Sihot ERP
- Inferenzmaschine Ontobroker zur Abbildung von Regeln bzgl. der Buchungsoptionen

03/2008 – 3/2010

**Eclipse Projekt SMILA und BROX eccenca  
Unified Information Access Architecture**

**Tätigkeiten**

- Konzeption und Architektur Bereich Storages.
- Entwicklungs-Prozess
  - Code Conventions
  - Eclipse Intellectual Property Process
  - Eclipse Build
  - Tests
  - Dokumentation
- Integration verschiedener Technologien
- Teamgröße: bis zu 15 (BROX, empolis und Offshore)

**Technologien**

- Eclipse
- OSGi/Equinox
- PDE Build
- p2 – Update

Die Firmen BROX und empolis haben beide ein hausinternes Such-Framework und entwickelt und vermarkten dieses in Zusammenhang mit einer eigenen Such-Engine. Um die Aufwände für die Wartung und Weiterentwicklung des Frameworks zu bündeln und neue Technologien einzubeziehen wurde beschlossen, auf Basis der Best-Practices beider Frameworks das neue Framework SMILA zu schreiben, dass den neueren Anforderungen in großen Unternehmen genügt und welches im Kern öffentlich ist. eccenca baut auf diesem Kern auf und erweitert diesen um BROX-eigene Komponenten. Der Anfängliche Schwerpunkt meiner Arbeit lag im Bereich der Storages insbesondere der Verwendung von XML Datenbanken. Darüber hinaus habe ich wesentliche Beiträge zur Definition des Entwicklungsprozesses und der Entwicklungsstruktur im Projekt geleistet und deren Übereinstimmungen mit den Anforderungen seitens Eclipse. Im weiteren Verlauf habe ich verschiedene suchrelevante Technologien untersucht und teilweise in das Framework integriert.

*Infos zu älteren Projekten kann ich aber bei Interesse nachreichen*

## Skill Matrix

Level: 1 = keine Ahnung, aber Begriff gehört, 10 = da gibt es nichts mehr dazuzulernen

Skill	Level	Jahre	Detailierung
<b>Methoden</b>			
Entwicklungsprozess (Agile Development)	7	Ab 2005	<p>Bis 2016 habe ich in keinem Projekt formal Agile Methodik genutzt aber auch nie ein reines V- oder Wasserfallmodell. Die Wahrheit liegt, wie so oft, dazwischen, d.h. viele der Agilen Prinzipien ergeben sich in der Praxis von selbst, will man eine funktionierende und für den Kunden nützliche und nutzbare Software erstellen, auch wenn formal ein eher klassischer Prozess definiert ist.</p> <p>Wesentlichster Punkt ist für mich als Teamlead, der meistens auch für Analyse Konzeption und Planung zuständig war, auf Änderungen von Anforderungen und Gegebenheiten reagieren und eingehen zu können. Das schmeißt zwar immer wieder Planungen über den Haufen, aber am Ende muss die Software die tatsächlichen Anforderungen des Kunden erfüllen und nicht unbedingt was zu Beginn gefordert wurde, selbst wenn es abgestimmt und spezifiziert war.</p> <p>Zum Glück werden seit einiger Zeit auch in Konzernen auf Agile Methoden formal umgestellt.</p>
Scrum	7	Ab 2016	<p>Ab 2016 wurde im Projekt Engpasstool auf Scrum umgestellt. Hierbei übernahm ich für die bislang verantwortete Applikation einen Gutteil der Rolle des Product Owners sowie die des Entwicklers (obwohl diese Rollen nicht in Personalunion ausgeführt werden sollten).</p>
Team- und Projektleitung	5	Bis 2016	<p>Kernaspekte sind hier aus der Entwicklung heraus</p> <ul style="list-style-type: none"> <li>• die Aufwände realistisch zu schätzen</li> <li>• in einen Release-Zyklus einzuplanen</li> <li>• die Aufgaben im Team zu verteilen</li> <li>• die Koordination mit anderen Bereichen bzw. Schnittstellen-Partnern</li> <li>• sich um das Ganze Drum-Herum (z.B. Entwicklungs-Infrastruktur, Berechtigungen und System-Zugängen, etc.) zu kümmern (Mädchen für alles)</li> </ul>

Skill	Level	Jahre	Detailierung
Anforderungs-Analyse Lösungs-Design und Spezifikation	8	Seit 2005	<p>Meine starke analytische Fähigkeit auch komplexe und komplizierte Sachverhalte zu durchdringen, sie zu formulieren und in verdaubarere Teile zerlegen zu können, sind für mich Schlüssel-Faktoren in diesem Bereich. Oft werden diese Tätigkeiten separat betrachtet und haben auch eigene Schwerpunkte und Ziele, aber sie komplett zu trennen (klassisches Modell) erachte ich nicht als sinnvoll, weil sehr große Wechselwirkungen bestehen und u.A. die Agilen Entwicklung motiviert haben. Wesentlich ist für mich hierbei das Formulieren und Verbalisieren sowohl der Anforderung als auch der zugehörigen Lösung(en) in einem konsistenten und wohl definierten Vokabular (DSL).</p> <p>Die zahlreichen Dokumente, die ich hier selber verfasst habe, reich(t)en von wenigen bis zu Werken mit jenseits von 100 Seiten. Als sehr hilfreiche empfinde ich es hierbei, mit dem breiten Wissen aus der Entwicklung zu praktikablen Lösungskonzepten zu kommen, die auch mit vertretbarem Aufwand implementiert werden können.</p>
Unit Testing/TDD	8	Ab 2005	<p>TDD nutze ich sehr gerne aber nicht in der reinen Form, da diese mir zu kleinteilig ist und der Code aus den vielen Zwischenschritten am Ende weggeworfen wird. Von den Auswirkungen schätze ich:</p> <ul style="list-style-type: none"> <li>• Fokus auf das Ziel, was die Auseinandersetzung mit den Anforderungen erzwingt</li> <li>• Den produktiven Code aus einer anderen Warte zu betrachten</li> <li>• testbarere Einheiten (DI)</li> <li>• eine bessere Testabdeckung <ul style="list-style-type: none"> <li>○ eine echte Kunst ist es die Testfälle möglichst orthogonal anzuordnen.</li> </ul> </li> </ul>
Mocking	7	Ab 2012	jMockit, EasyMock und Mockito,
Rollout- /Release- /Deploy -Management	5	5	Hierbei handelt es sich i.d.R. um die Installation der Entwicklungs-Pakete auf dem Server durch den Betrieb im Konzern. Dieses war entsprechend vorzubereiten und mit allen Beteiligten abzustimmen (z.B. Firewall-Freischaltungen für neue Schnittstellen, MQ Konfigurationen, etc.) aber es schließt auch Nutzerschulungen mit ein.

<b>Skill</b>	<b>Level</b>	<b>Jahre</b>	<b>Detailierung</b>
UML	4	5	Ehrlich gesagt: UML wird immer gefordert, tatsächlich habe ich es bisher kaum nutzen müssen, lediglich das ein oder andere State-, Activity- und Use Case-Diagramm und dann nur im Rahmen der Spezifikation bzw. Diskussion im Team.
Software/Applikations-Architektur	7	Ab 2010	Design der inneren Architektur der Applikation: d.h. Aufteilung der Module und Schichten, genutzte Technologien & Frameworks. Design der äußeren Architektur anhand der Rahmenbedingungen: d.h. Schnittstellen und Protokolle zu anderen Systemen, Integration in die Betriebslandschaft
Patterns	7	Seit 2007	Das schönste ist, wenn man ein Muster angewandt hat, ohne es vorher zu kennen.
<b>Softskills</b>			
Motivation	9		Ich hatte noch nie ein Problem mich zu motivieren, höchstens mit mangelnder Motivation anderer. Insofern ist auch meine Belastbarkeit eher überdurchschnittlich, denn „wenn es Freude macht, ist es keine Arbeit“; andere sagen dazu „Workaholic“
Arbeitsweise	8		<ul style="list-style-type: none"> <li>• eigenständige, strukturierte und nachvollziehbare Arbeitsweise</li> <li>• sehr Effizienz-orientiert</li> <li>• möglichst wenig Planungsoverhead</li> <li>• immer auf der Suche nach Optimierung/Automatisierung der Abläufe und Prozesse</li> </ul>
Kommunikation	7		Offene und zielgerichtete Kommunikation gegenüber allen Projektbeteiligten mit dem Fokus auf „Denotation“. Darüber hinaus ist mir natürlich auch die konnotative Dimension bewusst und ist u.A. wichtig, um frühzeitig auf Probleme in der Beziehungs-Ebene im Team reagieren zu können.
Teamfähigkeit	8		<ul style="list-style-type: none"> <li>• Bringe mich stets ein</li> <li>• Hilfe ohne Umschweife</li> <li>• Hatte bislang nur 1x ein nennenswertes Problem auf Beziehungsebene mit einem Kollegen und in dem Fall war ich nicht der einzige</li> <li>• Kann mich auch unterordnen</li> <li>• Äußere meine Meinung und Sicht, gehe aber nicht davon aus, dass sie übernommen werden muss</li> </ul>

<b>Programmierung</b>			
Java	8	Ab 1998	<p>Java 11</p> <ul style="list-style-type: none"> <li>• Ich warte auf Multi-line String literals.</li> <li>• Lambada Ausdrücke sind immer noch fantastisch!!!</li> </ul> <p>Von jeher ein Backend-Entwickler gewesen und darin liegt meine Stärke. Aufgrund des Bedarfs im Enterprise-Umfeld habe ich in den letzten Jahren vermehrt Webapplikationen entwickelt und kenn mich daher auch inzwischen recht gut mit Web-Frontend-Techniken aus.</p> <p>Bislang hatte ich kaum Projekte zur Entwicklung eines Fat-Clients, hätte aber großes Interesse an einer RCP Anwendung mitzuwirken.</p>
Spring Boot - Data - REST - Cloud - ...	6	ab 2018	Mit dem Schwenk zu Microservices kam auch Spring Boot, das ich nicht mehr missen möchte.
Spring	7	Ab 2010	<ul style="list-style-type: none"> <li>• In diversen Kundenprojekten als DI Container eingesetzt</li> <li>• „Der kleine Bruder von OSGi“</li> <li>• Aspekte <ul style="list-style-type: none"> <li>○ Profile</li> <li>○ AOP</li> <li>○ XML- und Annotationsbasierte Konfiguration</li> <li>○ Data-Integration</li> </ul> </li> </ul> <p>Unit-Testing</p>
J2EE	5	11	<p>Aufgrund der Kundenanforderungen in den Projekten hatte ich bis auf das Projekt bei der Talanx nie den Bedarf für einen vollen J2EE Server; ein Tomcat + Spring + ggf. JPA/JMS reichte aus.</p> <ul style="list-style-type: none"> <li>• Bei der Talanx wurde das Thema EJB aber weitestgehend in einem hauseigenen Framework versteckt und die Entwicklungsarbeiten war i.d.R. so weit davon entfernt, so dass dies nur wenig zum Tragen kam.</li> </ul>
AOP	5	3	<p>Bislang nur Spring AOP via Proxies und kein AspektJ (Bytecode-Level). Eigene Aspekte geschrieben im Projekt 2012 zur</p> <ul style="list-style-type: none"> <li>• generischen Fehler-Behandlung</li> <li>• Berechtigungs-Überprüfung</li> </ul>

JMS	6	6	<ul style="list-style-type: none"> <li>• Active MQ</li> <li>• IBM MQ Series</li> <li>• TIBCO EMS</li> <li>• Queues, Topics</li> <li>• Einfacher Austausch XML-basierter Nachrichten</li> </ul>
AMQP	4	Ab 2021	<ul style="list-style-type: none"> <li>• Rabbit MQ</li> </ul>
JPA	7	Ab 2007	<ul style="list-style-type: none"> <li>• Hibernate 6</li> <li>• Mapping der Entities auf DB Schema bzw. Generierung des Schemas aus den Entities heraus.</li> <li>• Custom und Embedded Types</li> <li>• Inheritance Mapping</li> <li>• Interceptors</li> </ul>
Eclipse Scout	3	Ab 2020	Bislang nur ein paar Implementierungen und Anpassungen von Maskenteilen
JSF 2	6	2012-2016	Im Rahmen meiner Kernrolle des Teamleiters und Architekten im VW Projekt in 2012 habe ich mich zunächst theoretisch mit den Grundlagen befasst und dann bei Problemstellungen auch praktisch Lösungen entwickelt und umgesetzt. Insofern besitze ich ein gutes Grundverständnis der Frameworks (Phasen, Komponenten, Zusammenspiel aller Teile) vor allem auf der Server-Seite.
Vaadin	3	2016	Mit Vaadin bin ich im TUI Projekt mit in Berührung gekommen und ist analog zu JSF ein Komponentenbasiertes Web-Framework. Im Rahmen der Tätigkeiten habe ich mich in ein paar Aspekte einarbeiten können, was erstaunlich leicht ging.
YAML	7	Ab 2018	<ul style="list-style-type: none"> <li>• Bevorzuge ich deutlich vor JSON für Konfigs</li> </ul>
JSON	9	Ab 2018	<ul style="list-style-type: none"> <li>• Inzwischen in einigen Projekten genutzt, meist mit Jackson o.Ä.</li> <li>• Im Vergleich zu XML und YAML fehlt mir hier immer wieder, dass es keine Kommentare gibt</li> <li>• Insofern OK für reinen Datatransport in der Applikation</li> </ul>
XML	8	Ab 2000	<ul style="list-style-type: none"> <li>• Neben dem normalen lesen/schreiben von XML Dokumenten verstehe ich darunter auch vor allem die Aspekte: <ul style="list-style-type: none"> <li>○ Validierung</li> </ul> </li> <li>• Generierung via JAXB</li> </ul>



XML Schema Design	8	Ab 2001	<ul style="list-style-type: none"> <li>• Habe in diversen Projekten XML Schemata entworfen, hauptsächlich mit XML Spy.</li> <li>• Im Grunde braucht man nur die gleichen Skills wie bei OO und DB Design <ul style="list-style-type: none"> <li>○ (multiple) Namespaces</li> <li>○ Ableitungen</li> <li>○ Groups</li> <li>○ Ref. Constraints</li> </ul> </li> <li>• ...</li> </ul>
XSLT	5	2001-2004	<ul style="list-style-type: none"> <li>• Normale Transformationen</li> <li>• Es muss bewusst sein: funktionales Paradigma</li> </ul>
Tomcat 7,8	6	10	<ul style="list-style-type: none"> <li>• Grundlegende Administrations-Kenntnisse zum Aufsetzen für die Entwicklung <ul style="list-style-type: none"> <li>○ Applikations-Kontexte</li> <li>○ Ports</li> <li>○ Bibliotheks-Management</li> <li>○ Rechte (LDAP Integration)</li> </ul> </li> <li>• SSL</li> <li>• Seit Spring Boot ist der Webserver idR integriert</li> </ul>
OSGi	5	2009-2012	<ul style="list-style-type: none"> <li>• Im Rahmen von SMILA/eccenca benötigt</li> <li>• Implementierung Equinox</li> <li>• Diese Welt ist riesig, weshalb ich mir hier nur Level 5 gebe. <ul style="list-style-type: none"> <li>○ Dependency Management</li> <li>○ OSGi Console</li> <li>○ Declarative Services</li> </ul> </li> <li>• Build (extrem aufwändig)</li> </ul>
Solr/Lucene/Elastic Search	7	2005-2012, 2016, 2018-2020	<ul style="list-style-type: none"> <li>• Hauptsächlich im Rahmen der Produktentwicklung der Brox GmbH als auch in diversen Kunden-Projekten eingesetzt.</li> <li>• Aspekte: <ul style="list-style-type: none"> <li>○ Importieren der Daten in den Index (~ETL)</li> <li>○ Design der Indizierung-Schritte und des Indexes, damit vernünftig gefunden werden kann</li> <li>○ Abbildung von Berechtigungs-Strukturen bei der Suche</li> </ul> </li> <li>• Integration von 3rd-Party Komponenten wie Thesauri und Wortstamm-Analyse/Reduktion</li> </ul>

<b>AWS Cloud</b>		2018-2021	Zum ersten Mal im TUI Projekt ab 2018 mit in Berührung gekommen (siehe dort)
Terraform	5	1	Ich hatte erst mit manuell Skripten unter Nutzung der AWS CLI begonnen. TF macht das Leben aber <b>viel</b> leichter!! <ul style="list-style-type: none"> <li>- Parametrisierung des Infra-Stacks so dass dieser für alle Stages verwendet werden konnte unter Nutzung von Workspaces</li> <li>-</li> </ul>
Beanstalk	4	1	Einfacher Weg skalierende Applikationen zu deployen aber halt etwas eingeschränkter
ECS	4	1	Der nächste Schritt nach Beanstalk seine Docker-Container zu deployen. Grundgerüst wurde bereitgestellt und ich habe hauptsächlich dieses dann an die Bedürfnisse angepasst.
ALB	4	1	Grundgerüst wurde bereitgestellt und ich habe gelegentlich die Regeln angepasst.
Cognito	4	2	Genutzt als einfaches User-Management für die UI mit manueller bzw. geskripteter Anlage der User
RDS/Aurora	4	2	Naja, es ist ein RDBMs in der Wolke, ohne nennenswerte Sonder-Konfigurationen.
SNS	3	2	Nur im Rahmen der AWS CodePipeline manuell eingerichtet Benachrichtigungen
CodeCommit	7	2	AWS Git Repo, mit viel Luft nach oben. Bin froh, dass wir auf Gitlab umgezogen sind.
CodeBuild, CodePipeline	6	2	Inzwischen gar nicht so schlechte Möglichkeit seine Applikationen zu bauen. Habe von Grund auf unsere CI/CD-Pipeline aufgesetzt
ParameterStore	7	2	Genutzt als Datenspeicher für die Parametrisierung der CI/CD Pipeline als auch der Applikation in den einzelnen Stages.
VPC	3	2	Genutzt zur Trennung der verschiedenen Stages auf Netzwerk-Ebene.
S3	4	2	Genutzt als ausschließlich interne Ablage für verschiedenste Artefakte mit Nutzung von Versionen.
X-Ray			Service "missbraucht" um im Betrieb eher fachliche Datenströme zu visualisieren
Kibana			ELK Stack für die Logs inkl. Einrichtung von Dashboard aber hauptsächlich genutzt um Logs für die Fehleranalyse zu sichten. Für komplexerer Analyse-Aufgaben habe ich dann aber eine kleines Tool geschrieben, das mittels Elasticsearch Logeinträge sucht und korreliert.
SecurityGroups	6	2	Genutzt um klassisch Zugriffe auf Ressourcen ähnlich zu einer Firewall zu beschränken.
IAM	5	2	Genutzt haben ich Rollen und Policies (auch Cross-Account) um weiter den Zugriff auf Ressourcen beschränken zu können.

Datenbanken			
Datenbank Design	7	Ab 2004	<p>DB Design ist eine eigene Disziplin und wenn auch eng verwandt mit OOD, so ist es doch hinreichend unterschiedlich dazu, um nicht ohne weiteres die Prinzipien von der einen in die andere Welt übertagen zu können: Es geht in der DB immer um Mengen und deren Relationen und nicht um Objekt-Graphen.</p> <ul style="list-style-type: none"> <li>• Aspekte <ul style="list-style-type: none"> <li>○ Indexe</li> <li>○ Constraints</li> <li>○ Trigger</li> <li>○ Locking</li> <li>○ Schemata</li> <li>○ Tabespaces</li> </ul> </li> </ul>
SQL	7	Ab 2000	<ul style="list-style-type: none"> <li>• Kein Projekt ohne auch eine DB zu haben und daher war auch immer SQL im Spiel, mal mehr mal weniger.</li> <li>• Wichtig hierbei ist das Denken in Mengen und Mengen-Operationen und nicht in einzelnen Schrittfolgen (Deklarative vs. Imperative Programmierung)</li> <li>• Vor allem in den DWH Projekten 2004-2007 habe ich enorm an Können gewonnen</li> </ul>
ETL	5	7	<ul style="list-style-type: none"> <li>• Kam in reiner Form in den DWH Projekten 2004-2007 vor als auch im Projekt Engasstool ab 2012</li> <li>• Aufgrund meiner Affinität zu Datenbanken und deren Operationen auf Mengen haben sich mir die hier nötigen Prinzipien und Handwerkszeuge schnell erschlossen und habe dem Thema ETL zunächst keine weitere Bedeutung zugemessen, weil in irgendeiner Form das immer in Projekten vorkommt – besonders in Konzernen.</li> <li>• Inzwischen habe ich gelernt, dass eine performante und vor allem wartbare (d.h. betrifft vor allem auch Fehlerdiagnose im Betrieb) von enormen Wert ist und derartige Schnittstellen zu designen auch erfahrene Entwickler vor eine große Herausforderung stellt.</li> </ul>

Microsoft SQL Server	6	8	<ul style="list-style-type: none"> <li>• Bezogen auf das MS spezifische Produkt: <ul style="list-style-type: none"> <li>○ Administration von DBs für Entwicklungszwecke</li> <li>○ Rechte</li> <li>○ Backup/Restore</li> </ul> </li> <li>• T-SQL <ul style="list-style-type: none"> <li>○ Hauptsächlich um Abläufe zu automatisieren oder zum Debugging</li> <li>○ Highlight: Umsetzung einer FSM im Rahmen eines DWH Projektes</li> </ul> </li> <li>• Tools <ul style="list-style-type: none"> <li>○ SQL Server Manager</li> <li>○ Query Analyzer</li> </ul> </li> </ul>
Oracle Database	4	5	<ul style="list-style-type: none"> <li>• Geringe Administrative Kenntnisse</li> <li>• PL/SQL <ul style="list-style-type: none"> <li>○ Aufwändigere STPs im Rahmen des Projektes Engpasstool ab 2012 für den ETL Prozess</li> </ul> </li> <li>• Tools <ul style="list-style-type: none"> <li>○ Oracle SQL Developer</li> <li>○ SQLPLUS</li> </ul> </li> </ul>
PostgreSQL	4	3	<ul style="list-style-type: none"> <li>• Einfache Administrative Kenntnisse: aufsetzen des Servers, import/export von DBs/Daten</li> <li>• Reine Nutzung als SQL DB via Standard-SQL</li> </ul>
Mongo DB 4	4	2021-2023	Einfache Queries und noch einfachere Aggregation Pipelines
<b>IDEs</b>			
IntelliJ	7	Ab 2021	Habe nach langer Zeit meinen Wiedersand dagegen aufgrund der Anforderungen in einem Projekt aufgegeben und finde es nun tatsächlich konsistenter als Eclipse, auch wenn ich von dort noch das ein oder andere Feature vermisste

Eclipse	8	Ab ~ 2004 - 2021	<ul style="list-style-type: none"> <li>• Überzeugter Nutzer von Eclipse trotz einiger Schwachstellen: <ul style="list-style-type: none"> <li>○ Egit – nutze daher die Git Bash</li> <li>○ Mangelhafter JS Support</li> </ul> </li> <li>• War bislang immer einer der Most-Advanced User in den jeweiligen Teams, bin mir aber sicher nicht alles zu wissen</li> <li>• J2EE und RCP Distributions</li> <li>• Plugins <ul style="list-style-type: none"> <li>○ Maven</li> <li>○ RBE</li> <li>○ MoreUnit</li> <li>○ JAutoDoc</li> <li>○ Spring IDE</li> <li>○ JRebel</li> <li>○ Mylyn</li> <li>○ AutoDeriv</li> <li>○ AsciiDoc</li> <li>○ PlantUML</li> <li>○ ...</li> </ul> </li> </ul>
<b>Microsoft Office</b>			
Word	9		<ul style="list-style-type: none"> <li>• Hauptsächlich und viel im Rahmen von Spezifikationen genutzt</li> <li>• Zur Automatisierung auch einige VBA Skripte erstellt</li> <li>• Styles sind selbst in der Version 2016 immer noch ein Kampf und die wenigsten wissen leider diese einzusetzen, was besonders schlimm ist, wenn diese dann Templates bereitstellen.</li> <li>• Seit 2016 schreibe ich nur noch wenig in Word, weil vielfach im Rahmen der Agilen Entwicklung zu Wikis übergegangen wurde. Dennoch bin in Word immer noch deutlich produktiver als in den Wikis und bevorzuge es daher.</li> </ul>
Excel	7		<ul style="list-style-type: none"> <li>• Übliche Office-Nutzung professionell und privat</li> <li>• Diagramme</li> <li>• Daten-Import/Export auch mit Java Bibliotheken</li> </ul>
PowerPoint	4		<ul style="list-style-type: none"> <li>• einfache Präsentationen</li> </ul>
OneNote	8		<ul style="list-style-type: none"> <li>• Ein super Wiki für die Dokumentation in kleinen Teams und auch probates Mittel gegen die persönliche Zettelwirtschaft</li> </ul>

<b>Betriebssysteme</b>			
WSL2	6	Ab 2020	Mit WSL2 habe ich nun für mein Entwicklungssetup die goldene Mitte zwischen der Produktivität der Windowsumgebung und den Linux Toolchains gefunden. D.h. meine IDE (via X-Server) und alles was für die Entwicklung nötig ist, läuft in der WSL, der Rest in Windows.
Windows	7		<ul style="list-style-type: none"> <li>Habe ich seit Anbeginn der Tage genutzt (Win 3.11)</li> <li>Der Level bezieht sich auf die Nutzung und nicht auf die Administration von Windows-Netzwerken, obwohl ich da auch ein paar Kenntnisse habe</li> <li>Nutze Windows viel über Scripting (.BAT) und über die Konsole</li> <li>Viele kleine Tools um das Leben zu erleichtern</li> </ul>
Linuxe	5		<ul style="list-style-type: none"> <li>Viel als Server innerhalb von Projekten genutzt aber nicht als OS zum Arbeiten; dazu bin ich einfach in Windows zu produktiv als dass ich die Not hätte, dass auf ein Linux zu übertragen.</li> <li>Von daher habe ich ausreichend Kenntnisse um ein Linux-Server/Docker-Image für die Entwicklung aufzusetzen, auch wenn dies oft noch mit viel googeln einhergeht.</li> </ul>
<b>Veraltet</b>			
C/C++	3	1	Im Rahmen meiner Studienarbeit und ein wenig Java JNI
VHDL	2	1	Im Rahmen meiner Diplom-Arbeit
C#	3	2	Konfigurations-FatClient
VB 6/VBA	3	3	Immer noch gut genug um schnell ein Makro in Office zu schreiben
Ant/Eclipse Build			
Perforce	3	<1	Nur kurzweilig im TUI Projekt in 2016 genutzt um es in den Grundzügen kennen zu lernen. Nach ein paar Monaten bin ich dann für meine lokale Arbeit auf Git/git-p4 umgestiegen.
Enterprise Architect	3	1	.. für das ein oder andere Diagramm
XML SPY	5	2000-2008	<ul style="list-style-type: none"> <li>Schema Design und Doku generieren</li> <li>XLST Entwicklung und Debugging</li> </ul>
Websphere 8	3	3	<ul style="list-style-type: none"> <li>Nur genutzt, weil vom Betrieb bereitgestellt</li> </ul> Marginale administrative Kenntnisse
ITIL	3	N/A	Habe hier den Grundkurs gemacht und bestanden, was für mich auf der Entwicklungsseite hinreichend ist, um die Prozesse im Application Support zu verstehen.