

Dennis Hoyer

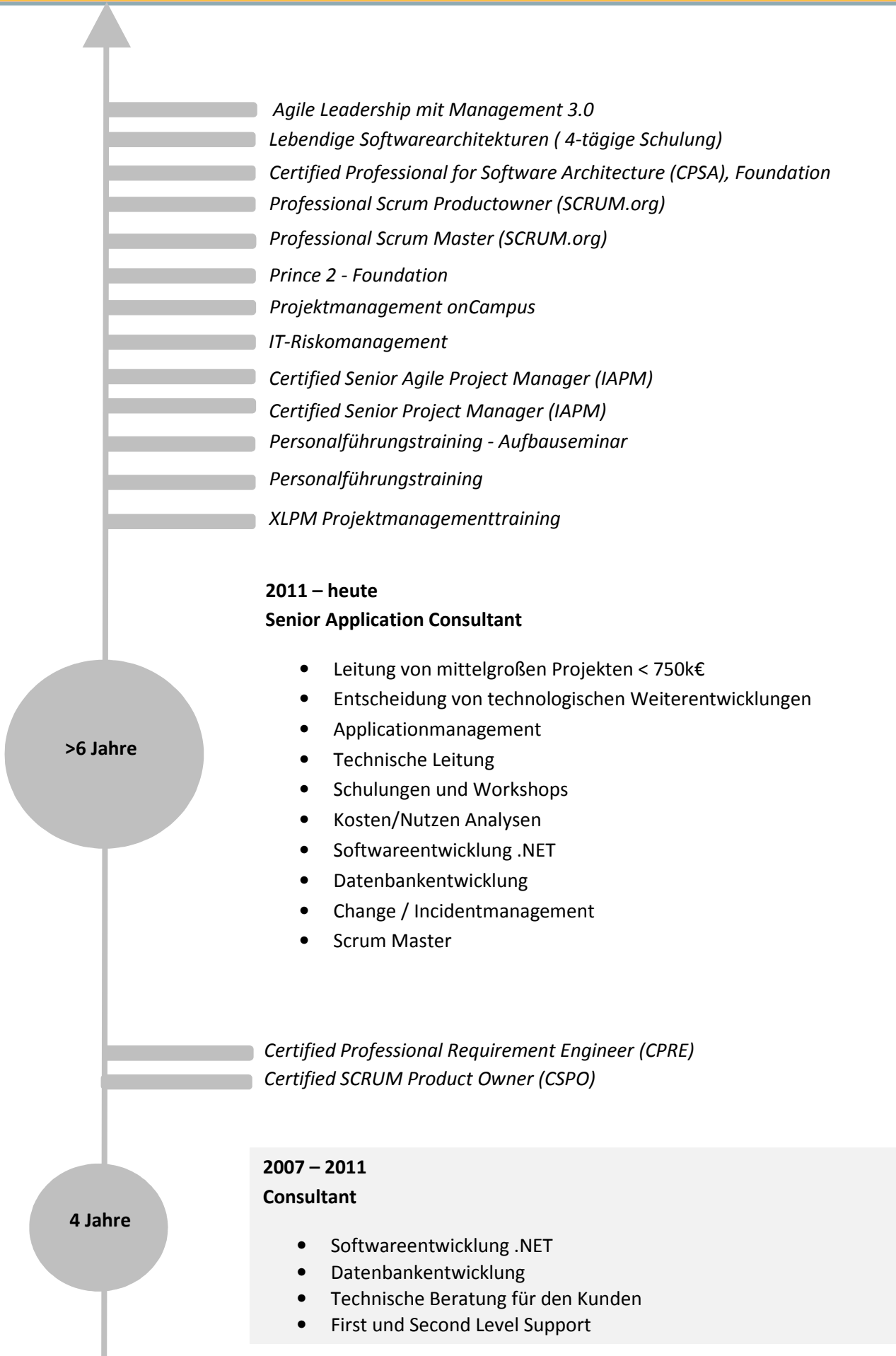


Dennis Hoyer ist seit 1998 in der IT Branche tätig. Zu seinen Schwerpunkten zählen die Softwareentwicklung, technische Beratung und das Applikation- sowie das Projektmanagement.

KARRIEREPFAD.....	2
SOFTWAREENTWICKLUNG.....	5
PROJEKTÜBERSICHT.....	10
PROJEKTMANAGEMENT.....	11
SUCCESS STORIES.....	14

KARRIEREPFAD







9 Jahre

1998 – 2007

Softwareengineer

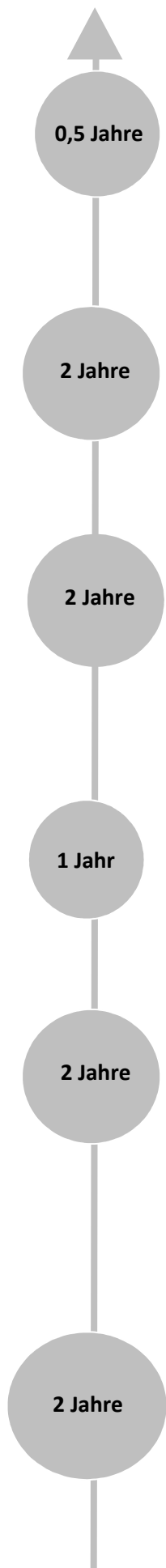
- Softwareentwicklung Centura, Delphi, C++, .NET
- Detenbankentwicklung
- Technische Beratung für den Kunden
- First und Second Level Support

Fernstudium Dipl. IT-Consultant (SGD)

Ausbildung IT-Fachinformatiker

SOFTWAREENTWICKLUNG





0,5 Jahre

2017-heute

Webservices

Entwicklung einer neuen Webanwendung auf Basis von Webservices via WCF

2 Jahre

2016-heute

CRM System / Kalkulator

Weiterentwicklungen im Bereich der Webanwendung.

Im Frontend wurde Sencha Ext JS verwendet und als Backend das .NET Framework

Es werden Weiterentwicklungen im Front sowie im Backend durchgeführt. Des weiteren werden Webservices via WCF an das System angebunden.

Steuerung externer und interner Mitarbeiter als Lead-Developer

2 Jahre

2016-heute

Preisinformationssystem

Entwicklung eines Microsoft WPF Clients zur Anzeige von Daten.

Der Client ist in .NET 4.5 entwickelt und setzt WPF, Dependency Injection und EF 6 ein. Die Software wurde zum Teil durch Dienstleister entwickelt und die Rolle besteht darin, den Client wartungsfähig umzubauen da dieser leider als FAT-Client umgesetzt worden ist.

1 Jahr

2016-2017

Folgeprojekt CRM System / Kalkulator

Weiterentwicklung des neuen CRM Systems / Kalkulator um weitere Vertriebsmöglichkeiten.

2 Jahre

2014-2016

CRM System / Kalkulator

System zur Berechnung und Speicherung von Kundendaten um für den deutschen B2B Bereich Energieverträge zu generieren.

Umsetzung erfolgt als JavaScript Frontend mit .NET Backend. Als Frontend Framework wurde das Sencha EXTJS 5 Framework gewählt. Im Zuge der einheitlichen Anforderungen und Bedienoberflächen wurden diese mit T4 Skripten aus dem Entity Framework generiert.

Die Kommunikation Frontend zu Backend wurde über WinAPI mit REST Services realisiert.

2 Jahre

2012 – 2014

Windows 7 Migration

Verantwortlich für die Umstellung von ~ 20 Anwendungen auf eine Windows 7 Umgebung, dabei enthalten unter anderem ein Siebel CRM System. Es musste hierbei die Tests und die Umsetzungen organisiert werden zusätzlich musste ein Projektvertrag aufgesetzt und die Budgets organisiert werden.

2 Jahre

2008 – 2011

Bundesweite Privatkunden Preisberechnung

Berechnung des Energiepreises für den Privatkundenvertrieb. Dabei werden verschiedene Schnittstellen zum SAP und Verivox verwendet.

In Phase 1 wurde die Entwicklung des ersten Programms zur Berechnung des Preises vorangetrieben.

In der Phase 2 wurden insbesondere Tests der Anwendung von mir durchgeführt, und dem Entwicklungsteam zur Abarbeitung berichtet. Dies hat maßgeblich zum Projekterfolg beigetragen. Des Weiteren wurde in der Phase 2 auch ein übergreifendes allgemeines Control entwickelt, welches dem Anwender ein sogenanntes Paging auf einem Grid ermöglicht.

Der Anwender kann dabei jeweils 50 Datensätze im Grid durchblättern. Das Grid basierte auf der Basisklasse von Infragistics Datagrid für Windows Forms.

Phase 3 beinhaltete die Entwicklung eines performanten Imports von Massendaten (>400.000 Datensätze) aus einer CSV Datei in eine Datenbanktabelle.

2011 – heute

Tool zum Importieren von Datensätzen in ein Siebel CRM System

Es wurde ein neues Projekt von einem Entwickler begonnen. Auf Grund des langsamen Fortschritts hat sich der Kunde entschieden, dass Herr Hoyer die Oberfläche erzeugen und sich in das Projekt einarbeiten soll.

Die bis zu diesem Zeitpunkt entwickelte Oberfläche (3 Wochen Entwicklungszeit) konnte nicht benutzt werden, da dort z.B. das Ado Entity Framework genutzt wurde, dies aber nicht dem Standard beim Kunden entspricht.

Der Client hat dabei direkt auf die Datenbank zugegriffen!

Auch das Binding war nicht korrekt und konnte nicht genutzt werden. Des Weiteren war auch das dahinterliegende MVVM Pattern nicht korrekt umgesetzt worden.

Das bedeutete das die Oberfläche von mir entkernt wurde, einzelne XAMLs konnten genutzt werden mussten aber auch umgeschrieben werden. Die einzelnen Schichten wurden über ein privat entwickeltes Modul extrahiert.

Innerhalb von 2 Tagen stand eine Oberfläche zur Verfügung die bereits den Durchstich zum Server hatte.

Nach weiteren 5 Tagen Entwicklungszeit war die Oberfläche nutzbar, die Basis war somit vorhanden. Der Durchstich zum Server war bis zu diesem Zeitpunkt in .NET Remoting gelöst. Da aber WCF genutzt werden soll, musste das Model mit den Facaden aufrufen entsprechend umgeschrieben werden. Innerhalb von 2 Stunden war dies realisiert.

Zum jetzigen Zeitpunkt werden bereits die BusinessLogiken im ViewModel umgesetzt, insgesamt ist durch den Entwicklungsschub der Zieltermin evtl. zu erreichen.

4 Jahre

2008 – 2010

Prognoseermittlung

Schnittstelle zu einem externen System, welches Prognoseberechnungen durchführt. Die Schwierigkeit in diesem Projekt liegt nicht in der Entwicklung sondern in der Fachlichkeit.

Es sind laufend Nachforschungen von Daten zu erstellen, um möglich Prozessrisiken im Unternehmen des Kunden zu minimieren. Die Kommunikation ist dabei zwischen den unterschiedlichen Abteilungen sehr wichtig, außerdem ist ein tiefgreifendes Verständnis der Daten erforderlich.

Die Entwicklung der Schnittstelle ist in mehreren Entwicklungsphasen unterteilt, der Betrieb (Nachforschungen) fallen jedoch immer an.

1 Jahre

0,1 Jahre

2010 – 2010

Migration auf 64-bit

Die bestehenden Systeme sollten von 32 Bit auf 64 Bit Systeme umgestellt werden. In der Vergangenheit gab es immer große Probleme mit der Speicher-Allokierung bei großen Datenmengen.

Die Problematik bestand nicht bei den .NET Modulen, diese basierten alle auf Individual-Entwicklung, jedoch hat eine zugekaufte Komponente Mehraufwand bedeutet. Es handelte sich dabei um die Komponente Asses. Da in der Vergangenheit nicht auf die neuesten Versionen gewechselt wurde, musste hier ein Versionsprung von Version 3.0 auf 9.2 vollzogen werden.

Dies wurde letztlich geschafft, nach ca. 2 Wochen Aufwand. Probleme gab es bei der neuen Klassenarchitektur und dem Umgang mit Word und PDF Dokumenten. Hier wurden einige Konzepte grundlegend von dem Herstellen der Software verändert. Hauptaufgabe bestand darin, für den Anwender den Umstieg möglichst ohne späteren Mehraufwand umzusetzen. Das wurde durch interne Konvertierungsmethoden realisiert, der Anwender hat keinen Mehraufwand.

0,5 Jahre

2009 – 2010

Verwalten von Strom-Zertifikaten

Das Projekt dient zur Optimierung von Bereichsübergreifenden Aufgaben zur Zertifikatsverwaltung von Stromzertifikaten. Hier wurde die gleiche Client und Server Architektur angewandt, wie bereits bei Projekt (Prozessoptimierung zwischen Vertrieb und BackOffice).

>5 Jahre

2009 – heute

Prozessoptimierung zwischen Vertrieb und BackOffice

Das System dient zur medienbruchfreien Kommunikation und löst eine Excel Vorlage ab. Bisher wurden die Daten zwischen den verschiedenen Stellen per Papier / Email / Excel vorlagen an die verschiedenen Stellen weitergeleitet. Dies wird mit dem neuen System abgeschafft und in einer WPF Anwendung benutzerfreundlich dargestellt. Der Vertriebler sieht den gesamten Prozess in einer sicheren Statusführung und kann erkennen ob an einer Stelle ein Engpass entsteht. Das System wird mit dem Vertriebsunterstützenden Kalkulationssystem zusammenbetrieben. Die Serverseitige Abbildung wurde in 3 Schichten durchgeführt. (siehe Vertriebsunterstützendes Kalkulationssystem) Bei diesem Projekt wurden drei Schichten auf der Client Seite abgebildet, dies wurde aus der Erfahrung bei allen WPF (Windows Presentation Foundation) Projekten durchgeführt. Dabei wurden folgende drei Schichten abgebildet.

Model

Kommunikation mit dem Server, bildet einzelne Datenbank Objekte ab

Viewmodel

Kapselt ein Model. Jedes Viewmodel kann dabei eigene Logik haben.

View

WPF Formulare.

Durch die Trennung in View, Viewmodel und Model ist es später möglich auch den Client durch NUnit z.B. zu testen. Im Projekt war es erforderlich sowohl fachlich als auch technisch beratend zur Seite zu stehen. Technisch wurden auch weitere Mitarbeiter im Projekt an die 3 Schichten Vorgehensweise und in der WPF Technologie geschult. Die Datenbank wurde auch inhaltlich erweitert und konzeptioniert.

0,5 Jahre

2006 – 2007

Anpassung des Opensource Projektes dotProject

Zunächst wurde eine Optimierung der Projektabläufe innerhalb der Unternehmung vorgeschlagen. Dieser Vorschlag wurde angenommen. Es wurde in einem Auswahlverfahren das OpenSource Projekt dotProject vorgeschlagen, um den Prozess softwareseitig zu unterstützen. Bei dotProject handelt es sich um ein Projektmanagementtool. Dieses wurde angepasst und Fehler beseitigt. Es traten insbesondere Fehler bei dem Zugriff auf die MySQL Datenbank Version 5 auf. Die Software wurde für den Zugriff auf eine niedrigere Version 4 geschrieben, jedoch wurden bei der neuesten Version erhebliche Änderungen vorgenommen. Dies hatte zur Folge, dass eine Anpassung der Software erfolgen musste. Zusätzlich wurden Berichte angepasst, Beispielsweise die Darstellung eines Gantt Diagrammes. Die Software wurde eingerichtet und installiert. Dabei wurden folgende Komponenten installiert. XAMPP beinhaltet Apache, MySQL, PHP, Mail, dotProject, etc. Diese wurden entsprechend auf dem Server unter Windows 2003 Server installiert.

4 Jahre

2003 – 2007

internationales Auditsystem (Qualitätsmanagementsystem)

Entwicklung eines Auditier Systems für einen weltweiten Automobilhersteller. Das System wurde für 2 verschiedene Systeme entwickelt. Zum einen für Windows CE Geräte (PDAs) zum anderen für Windows 2000 und höher. Der Benutzer ist fähig sowohl Daten auf dem PC als auch auf dem PDA zu erfassen und diese über das PC System in die Datenbank einzulesen. Das System ist zum Teil dezentral, da auch Zulieferer des Automobilherstellers, die keinen Zugriff auf das interne Netzwerk haben, die Daten in das System erfassen sollen. Somit musste eine eigene Replikation konzeptioniert und realisiert werden. Des Weiteren wurde auch eine Sprachenunabhängigkeit realisiert, da es wie oben beschrieben weltweit eingesetzt wird. Im Verlauf des Projektes wurde eine Intranetseite mit ASP.NET realisiert und ein IIS 6 mit .NET Unterstützung aufgesetzt. Das System wurde zunächst für den SQL Server entwickelt, wurde aber auch später für Oracle Server zur Verfügung gestellt. Für die Betreuung des Systems war es nötig Englisch zu kommunizieren, in Sprache und Schrift. Während der Betreuung und der Entwicklung des Systems wurden Qualitätsmanagementkenntnisse angeeignet und versucht verschiedene unterstützende Qualitätsmanagement Systeme einzusetzen (NUnit, Dokumentenmanagement, Human Resource Management System, BugTracking System, sowie ein Projektmanagementsystem)

2 Jahre

2001 – 2002

Knowledge-Datenbanksystem

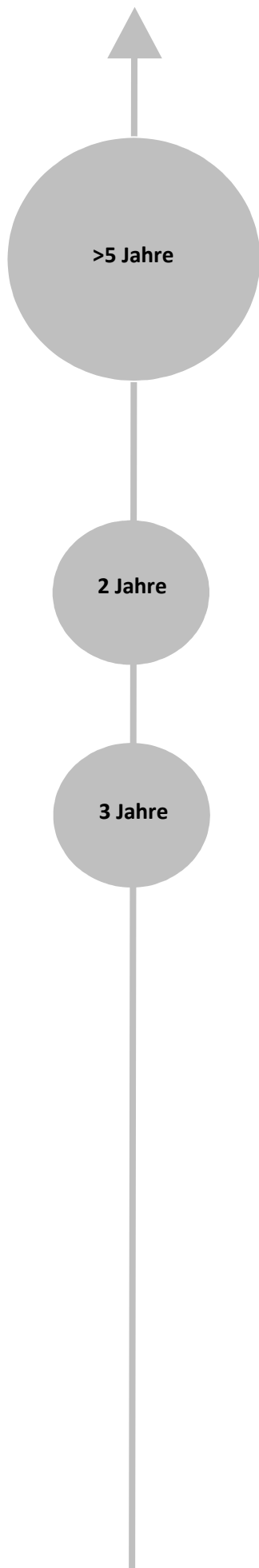
Umsetzung eines Pflichtenheftes für einen Namenhaften Flugzeugbauer. In dem System werden Probleme erfasst, welche später recherchiert und ausgewertet werden können. Die Anwendung wurde in VB6 programmiert und greift über die ADO Schnittstelle auf die Datenbank zu. Als Backenddatenbanksystem diente die Jet 4.0 Access Datenbank.

1 Jahre

2002 – 2002

3D Unfallkonstruktion

Neuentwicklung eines bereits existierenden Produktes. Für das neue Softwareprojekt wurde ein neues Konzept erstellt. Dabei wurde der bisherige Funktionale Aufbau in Klassen umgesetzt. Es wurden Klassen geschaffen die selbständig agieren können und nicht mehrfach existierenden Code benötigen. Bei der grafischen Umsetzung wurde auf OPEN GL zurückgegriffen. Die eigentlichen Daten wurden im XML Format abgelegt, die grafischen Objekte wurden im X3D Format abgelegt. Eine Zwischenstufe des Systems benutze auch das VRML Datenformat.



2007 – heute

Vertriebsunterstützendes Kalkulationssystem

Konzernweites Vertriebsunterstützendes Kalkulationssystem mit Anbindung verschiedenster Systeme, Bsp. CRM Systeme, Oracle Datenbanken. Bei diesem System handelt es sich um ein System zur Unterstützung der Berechnung des Strompreises. Des Weiteren wurde ein CRM System angebunden aus dem das System gestartet werden kann. Die Anwendung ist als verteilte Anwendung konzipiert und in 3 Schichten abgebildet. Das gesamte System ist in mehrere Satelliten aufgeteilt. Die Kommunikation wird über die .NET Remoting Technologie gesteuert

2000 – 2001

Navigationssysteme

In Zusammenarbeit mit einem Namenhaften Kartensystemhersteller wurde eine Software entwickelt, für die Logistikbranche. Diese Software stellt Fahrzeuge, welche mit einem Navigationssystem ausgerüstet sind, auf einer Karte da, somit ist die Logistik des Speditionsbetriebs jederzeit über den Standort der Fahrzeuge informiert.

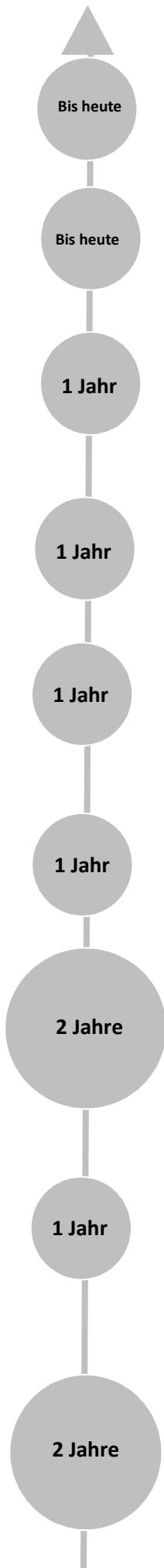
1998 – 2001

Lagerlogistik

Entwicklung einer Softwarelösung für den Werkstattbereich eines Speditionsbetriebs. Dabei wurde die Lagerhaltung der einzelnen Werkstoffe, sowie die Rechnungserstellung, Angebotserstellung abgebildet.

PROJEKTMANAGEMENT





2018 – heute
Preisinformationssystem

Scrum Master

2016 – heute
CRM System / Kalkulator

Application Manager, Agiler Transformator, Scrum Master

2016 – 2016
Folgeprojekt CRM System / Kalkulator

Projektleiter (> 300k€)

2015 – 2015
CRM System / Kalkulator

Projektleiter (> 650k€), Firejumper

2014 – 2015
Windows 7 Migration für die BusinessUnit

Projektleiter

2013 – 2014
Neues Softwaresystem zum Import von Massendaten

Arbeitspaketverantwortlicher

2013 – 2014
Windows 7 Migration

Arbeitspaketverantwortlicher

2013 – 2014
Einführung eines neuen Sales Produkts

Projektleiter

2012 – 2013
Einführung eines neuen CRM Systems

Arbeitspaketverantwortlicher



2003 – 2007
internationales Auditsystem

Technische Projektleitung

SUCCESS STORIES



Einführung agiles Projektmanagement

Am Anfang gab es lange Release Zyklen, die damit verbunden waren das die Produktqualität nicht gestimmt hat. Jedes Quartal die gleichen Fragen bei Beginn der Tests seitens der Fachbereiche,

- Was müssen wir testen ?
- Was haben wir geändert ?

und die Folge, dass letztlich so viele Änderungen an dem System waren, dass die Auswirkungen unbekannt waren und viele Fehler produktiv gingen.

Anfang des Jahres (2018) haben wir IT Seitig den Entwicklungsprozess im Zuge der CIO Entscheidung auf Agil eigenverantwortlich umgestellt.

Anfang war sehr viel Skepsis auf der fachlichen Seite, jedoch hat sich dieser Gedanke schnell geändert.

Es wurden mittlerweile 12 Sprints vollzogen, es gibt keine Incidents mehr und auch keine betrieblichen Ausfälle. Früher wurden ~110 Anforderungen live geschaltet, jetzt sind es rund 13 alle 3 Wochen.

Dabei gibt es alle 3 Wochen ein Planning, wobei die IT die Technical Depths vorgibt die umgesetzt werden müssen, das bedeutet, diese werden seitens IT vorab eingeplant. Der fachliche Produktowner plant dann die restlichen Verfügbaren Ressourcen. So bleibt die Applikation IT-seitig in einem betriebsarmen Zustand und es schafft auch Zeiträume für nötige Arbeiten auf der technischen Seite.

Täglich gibt es ein Daily in dem die Entwickler sich austauschen können, auch hier ist das Business optional eingeladen um eine größtmögliche Transparenz zu schaffen.

Das Backlog wird indes vom Fachbereich oder von der IT gefüllt, das Refinement finden wöchentlich einmal statt.

Das Ergebnis ist ein transparenter, erfolgreicher Prozess, der sowohl das Business als auch die IT zufriedenstellt und die Produktivität bestmöglich ausschöpft.

Rückblickend war die größte Herausforderung die Projektkultur zu verändern. Am Anfang war die Angst vor Verlusten, da es hieß, dass wir statt Quartalsweise jetzt 3 Wöchig ausrollen. Pro Quartal sind es also ungefähr 4 Rollouttermine mit jeweils ~13 Tickets. Natürlich sind es im ersten Blick weniger Anforderungen, aber dafür die richtigen und die in der Timebox von 3 Wochen testbaren Anforderungen.

Auch die Skepsis vor den verkürzten Testzeiträumen war sehr ausgeprägt bei der Vorstellung wie der agile Prozess laufen könnte, dies wurde aber schnell ausgeräumt nachdem festgestellt worden ist, dass ~13 Anforderung wesentlich einfacher zu testen sind als 100erte.

Letztlich waren die Schlagwörter Transparenz, Testbarkeit, Risikominimierung, Testressourcen die halfen, die "Kritiker" zu überzeugen.

Der Scrum Master Rolle war hierbei sehr wichtig, zum einen zum Coachen des Fachbereich aber auch auf der IT Seite. Anforderungen die zu groß sind, müssen gesplittet werden um umsetzbar und testbar zu bleiben. Dabei sind die IT Kollegen bei diesem Splitting zu unterstützen den auch auf dieser Seite kann man schnell wieder in alte Muster verfallen.

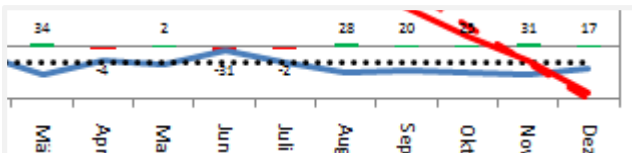
Unterstützt wurde das agile Projektmanagement mit der JIRA Software, die die Anforderungen als Tickets abbildet und einen festen Workflow bietet.

```
<Global.System.Diagnostics.DebuggerNonUserCode  
Global.System.CodeDom.Compiler.GeneratedCode  
Public Function FindByID(ByVal ID As String)
```

Softwareentwicklung

Durch die verschiedene Projekte und Einsatzbereiche wurden verschiedenste Programmiersprachen und -techniken eingesetzt.

- .NET (C#, VB.NET)
- Windows Forms; WPF
- WCF, WinAPI, RestServices, .NET Remoting
- Delphi, C++, VBA
- JavaScript; Sencha Ext JS
- T4
- Dependency Injection



Analysen und Reporting

Mit dem umfassenden Datenbank Knowhow ist Herr Hoyer auch in den Bereichen Analysen von Datenbeständen sowohl technisch als auch fachlich einbezogen worden.

- Analyse von großen Datenbeständen
- Technische und fachliche Beratung zu Analyseergebnissen

```
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STAT  
STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FI
```

Datenbanken

In nationalen und internationalen Einsätzen wurden verschiedenste Datenbanksysteme eingesetzt, so z.B.

Microsoft SQL Server, Oracle.

Seit 1998 hat Herr Hoyer ständig sein Knowhow weiteraufgebaut

- SQL
- Microsoft SQL Server
- ORACLE
- GUPTA, Interphase, MS Access
- Replikationen, Trigger, Index, Execution Pläne

Windows Forms ?

C++ ?

WCF ?
WPF ?

VB ?
C# ?

Consulting / Beratung

Ende 2007 ist Herr Hoyer aktiv in den Bereich Consulting gewechselt. Durch seine schnelle Auffassungsgabe konnte er sich stets in die Vorort befindlichen Projektteams einarbeiten. Schnell hat er seine Tätigkeit als Consultant aufgenommen und folgende Rollen wahrgenommen.

- Projektleitung
- Application Management
- Softwareentwicklung
- Tagesgeschäftberatung
- Unterstützung bei technologischen Entscheidungen
- Leitung externer Berater
- Betriebs- und Budgetüberwachung

Klassisch CPRE CSPO
IAPM AGIL

Projektmanagement

Herr Hoyer besitzt umfangreiches Projektmanagement-Wissen und ist bereits bei der IAPM zertifiziert.

- Certified Senior Project Manager IAPM
- Certified Senior Agile Project Manager IAPM
- Certified Professional Requirement Engineer
- Certified Scrum Product Owner (Scrum Alliance)
- Personalführungstrainings
- Dipl. IT Consultant (SGD)
- IT-Risikomanagement
- Prince 2 – Foundation
- Professional Scrum Product Owner (SCRUM.org)
- Professional Scrum Master (SCRUM.org)

ITIL v3

Servicemanagement

Herr Hoyer ist ITIL Zertifiziert.

- ITIL v3 Zertifizierung
- Incident Management
- Release Management
- Problem Management
- Remedy
- Redmine
- JIRA



Kontakt:

Dennis Hoyer

Kontakt@dennis-hoyer.com