

MOTIVATION UND DEFINITION

Das Kernelement der Wirtschaft von morgen ist die „Intelligente Vernetzung“

– Dr. Reinhold Festge in Digitale Strategie 2025, BMWi 2016

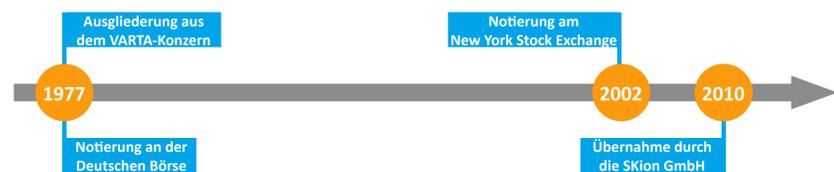
Der digitale Wandel beschreibt die Übertragung des Menschen und seiner Arbeitswelt auf die digitale Ebene. Dies sorgt für enorme Umwälzungen in Wirtschaft und Gesellschaft, bei Arbeit, Konsum, Kooperation und Kommunikation. Werden bestehende Geschäftsmodelle **intelligent** und **vernetzt** aufgestellt, spricht man vom *Connected Business*.

Auch im IT-Projektmanagement der ALTANA AG muss auf die anstehenden Veränderungen reagiert werden. Es gibt verschiedene neue Methoden und Technologien, die hier eingesetzt werden können, um die Effizienz und somit auch den Erfolg von Projekten zu steigern.

Ziel dieser Arbeit ist es, das IT-Projektmanagement zu untersuchen und Verbesserungsmöglichkeiten in den Bereichen Technik, Prozesse und Kommunikation sowie Daten im Kontext des digitalen Wandels in Form von Handlungsempfehlungen darzustellen.

ALTANA AG

Bei der ALTANA AG handelt es sich um ein global führendes Unternehmen der Spezial-Chemie mit Hauptsitz in Wesel. Die ALTANA AG unterhält weltweit 47 Produktions- und 50 Service und Forschungslabor-Standorte und beschäftigt rund 6.000 Mitarbeiter (Stand 2017).



IT-PROJEKTMANAGEMENT BEI DER ALTANA AG

Bei der ALTANA AG spricht man von IT-Projekten, wenn **IT-Systeme**, **IT-Prozesse** oder **IT-Mitarbeiter** betroffen sind und der **interne IT-Aufwand** ≥ 10 **Manntage** ist oder die **internen + externen IT-Kosten** ≥ 10.000 € liegen. Trifft dies nicht zu, handelt es sich um ein *Change Request*.

Derzeit gibt es bei der ALTANA AG ca. **170 laufende IT-Projekte**.

Aktuell wird ein selbst-entwickeltes Projektmanagementsystem aus folgenden Komponenten verwendet:



METHODEN UND ANALYSEN

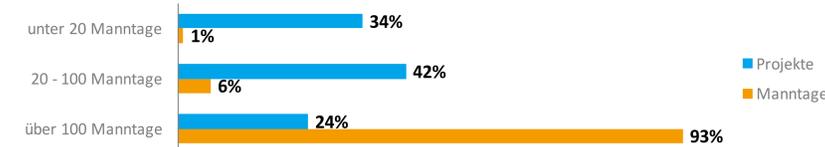
Zur Analyse des IT-Projektmanagements bei der ALTANA AG wurden die folgenden Methoden durchgeführt:

- Globale Projektstrukturanalyse (591 Projekte seit 2005)
- Empirische Expertenbefragung der IT-Projektleiter
- CMMI-Bewertung des IT-Projektmanagements

Die Ergebnisse dieser Analysen werden im folgenden dargestellt.

METHODE 1 GLOBALE PROJEKTSTRUKTURANALYSE

Geplanter Aufwand



Benötigte Dauer

- 2% der Projekte benötigen **weniger als 20 Kalendertage**
- 10% der Projekte benötigen **20 bis 100 Kalendertage**
- 88% der Projekte benötigen **über 100 Kalendertage**

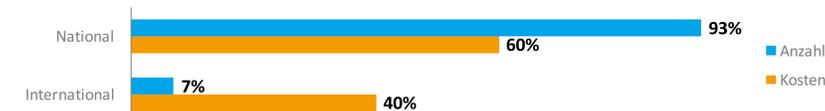
Kosten

- 92% der Projekte benötigen **weniger als 100.000 €**
- 7% der Projekte benötigen **100.000 – 1.000.000 €**
- 1% der Projekte benötigen **mehr als 1.000.000 €**

Teamgröße

- 24% der Projekte benötigen **einen internen Projektbeteiligten**
- 47% der Projekte benötigen **2 bis 5 interne Projektbeteiligte**
- 29% der Projekte benötigen **mehr als 5 interne Projektbeteiligte**

Internationalität



METHODE 2 EMPIRISCHE EXPERTENBEFRAGUNG

Im Rahmen einer qualitativen Befragung wurden 16 IT-Projektleiter zu ihrer Zufriedenheit mit dem IT-Projektmanagement befragt. Da die Umfrage zu einem bestimmten Zeitpunkt durchgeführt wurde, weist sie das Forschungsdesign einer Querschnittsstudie auf. Die Umfrage wurde mit Hilfe von Google Forms durchgeführt und beinhaltet 29 Fragen.

Die wichtigsten Erkenntnisse sind folgende:

- 37,5% der Projektleiter sind **unzufrieden** mit dem Projektmanagement
- 50% der Projektleiter sind **unzufrieden** mit der Kommunikation in Teams
- 50% der Projektleiter bewerten das aktuelle Projektmanagementsystem mit **befriedigend**

METHODE 3 CMMI-BEWERTUNG

Bei CMMI (*Capability Maturity Model Integration*) handelt es sich um ein Referenzmodell des *Software Engineering Institute* der *Carnegie Mellon University*, welches zur Beurteilung von Prozessen verwendet wird. Im Rahmen der Bachelorarbeit wurden vier Prozessgebiete des Projektmanagements untersucht:



LÖSUNGSANSÄTZE

Zusätzlich zu den Ergebnissen der durchgeführten Analysen wurden die Ergebnisse der ALTANA Marktforschung 2016, sowie Trends der Digitalisierung aus der verwendeten Literatur zum Erstellen der Lösungsansätze hinzugezogen.

Technik

Ich empfehle die Verwendung einer standardisierten Projektmanagementplattform als *Software as a Service* anstelle des aktuell verwendeten Systems (*SharePoint, Project, Excel*).

Vorteile:

- Skalierbarkeit
- Kollaborations-Tools
- API
- Mobile App
- Einbindung von Bots
- keine Anschaffungskosten

Prozesse und Kommunikation

- Einführung agiler Methoden (z.B. SCRUM)
- Einführung einer neuer Arbeitsorganisation („Arbeit 4.0“, BMAS 2016)
 - ⇒ Externe Flexibilisierung (Outsourcing, Leiharbeit)
 - ⇒ Interne Flexibilisierung (flexiblere Arbeitszeitmodelle, agile Arbeitsformen)
 - ⇒ Räumliche Dezentralisierung und Virtuelle Teams

Daten

- Datensicherheit
- Vermeidung von Datensilos und doppelter Datenhaltung
- Big Data / Predictive Analytics

HANDLUNGSEMPFEHLUNG

Aufgrund der beschriebenen Lösungsansätze stelle ich folgende Handlungsempfehlung aus.

Projektmanagementplattform als *Software as a Service*

Aufgrund einer durchgeführten Marktrecherche mit einem entwickelten Anforderungskatalog empfehle ich folgende Systeme:



Kollaborations-Tool / Social Intranet

Im Folgenden werden einige etablierte Kollaborations-Tools beziehungsweise Social-Intranet-Lösungen aufgezählt, ohne eine konkrete Empfehlung für die ALTANA AG abzugeben.



FAZIT

Im empfehle der ALTANA AG folgende Roadmap zum *Connected Business*:

- Einführung einer neuen standardisierten Projektmanagementplattform, die als *Software as a Service* eingesetzt wird und eine globale und mobile Nutzung ermöglicht
- Einführung einer Kollaborationsplattform, die die Kommunikation in virtuellen Teams verbessert
- Nutzung von agilen Methoden, die die Effizienz und Motivation der Mitarbeiter in Projekten verbessern

Implementierung eines Visualisierungskonzeptes

Darstellung von Komponentenstruktur und Qualitätsmetriken in der Softwareanalyse

Motivation

Die Wartung von Software macht einen häufig unterschätzten Zeit- und Kostenanteil aus:

- 85-90%** des Budgets werden in **Betrieb und Wartung** von Altsystemen investiert. [1]
- 50%** ihrer Zeit investieren Entwickler, um Quelltext **nachzuvollziehen**. [2]

Umfragen zeigen, dass **Softwarevisualisierung** dabei eine geeignete Methode darstellt, um diesen Zeitaufwand und die damit einhergehenden Kosten zu verringern.

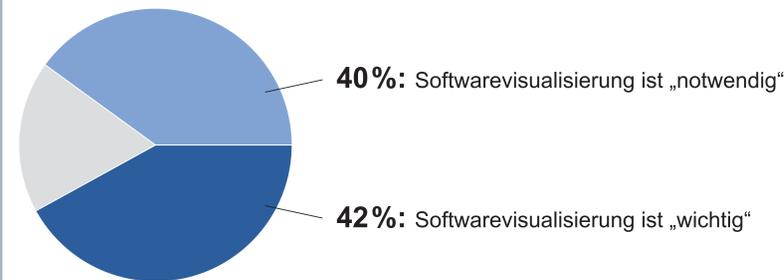


Abbildung 1: Die Rolle der Softwarevisualisierung [3]

Gründe für Softwarevisualisierung [4]:

1. Zeit- und Kostenersparnis
2. Besseres Verständnis von Software
3. Steigerung der Produktivität und Qualität

Ziel der Arbeit: Entwicklung eines Konzeptes zur Visualisierung der Struktur und Qualität von Software und Implementierung in Form einer prototypischen Webanwendung.

Zielgruppe: Entwickler oder Leiter eines Software-Projektes können sich einen Gesamteindruck des Systems verschaffen, mögliche Schwachstellen aufdecken und entsprechende Gegenmaßnahmen einleiten.

Softwareanalyse

Die Ergebnisse einer **statischen und evolutionären Softwareanalyse** bilden die Basis für die Visualisierung. Anhand von Metriken können dabei verschiedene Eigenschaften von Software quantifiziert und ausgewertet werden. In Kombination mit Informationen aus einer Versionsverwaltung können diese Metriken in einem **zeitlichen Verlauf** dargestellt werden.

Für die Analyse kommt die Anwendung „Coderadar“ zum Einsatz. Sie bietet die Möglichkeit, beliebige Java-Projekte zu analysieren und speichert die Ergebnisse dieser Analyse in einer Datenbank. Über eine REST-Schnittstelle können die gespeicherten Daten abgefragt werden.

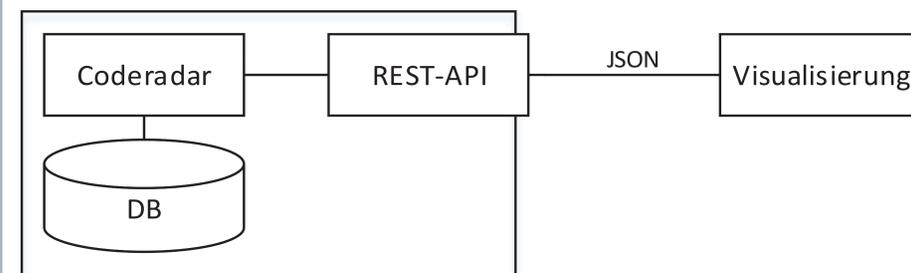


Abbildung 2: Aufbau der Coderadar-Schnittstelle

Visualisierungskonzept

Aspekte einer guten Visualisierung:

- Verständliche Darstellung
- Ver- und Entschlüsseln der visualisierten Daten
- Keine Verfälschung der originalen Daten
- Exploratives Erkunden

In der Informatik werden **Metaphern** eingesetzt, um **komplexe Strukturen und Zusammenhänge einfacher zu verstehen**. Ein oft behandeltes Konzept zur Visualisierung von Software nutzt die Metapher einer Stadt: Klassen werden als Gebäude, Module als Stadtteile dargestellt.



Abbildung 3: Visualisierung von Struktur und Komponenten

An dieses Konzept wird im Rahmen der entwickelten Anwendung angeknüpft und es werden Aspekte zur **Darstellung der Veränderung** zwischen zwei Versionen ergänzt.

Qualitätsmetriken

Die Qualitätsmetriken werden dabei auf **drei verschiedene Eigenschaften** der Quader abgebildet. Welche Eigenschaft welche Metrik widerspiegelt, kann frei gewählt werden.

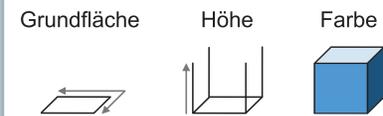


Abbildung 4: Abbildung von Metriken auf die Eigenschaften der Quader

Komponentenstruktur

Eine optimale und platzsparende Anordnung der Grundflächen aller Komponenten wird mit Hilfe des **Bin Packing-Algorithmus** berechnet:

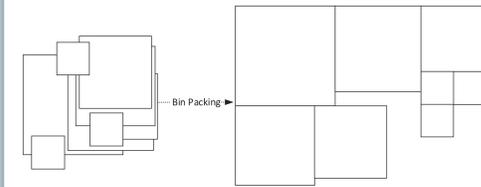


Abbildung 5: Anwendung des Bin Packing-Algorithmus

Evolution

Um die Veränderung der Software **zwischen zwei verschiedenen Zeitpunkten** und die resultierenden Unterschiede zu visualisieren, können folgende Arten einer Veränderung dargestellt werden:

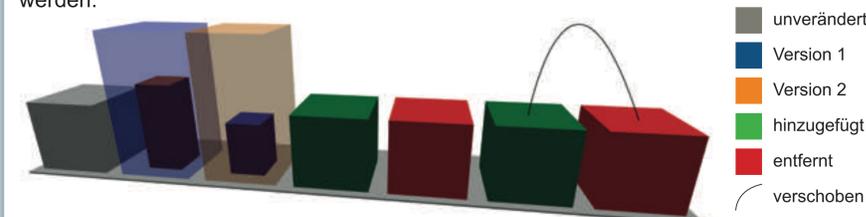


Abbildung 6: Visualisierung der Veränderung zwischen zwei Versionen

Implementierung

Da die Anwendung zukünftig in ein webbasiertes Dashboard integriert werden soll, wird sie ebenfalls mit **Webtechnologien** (HTML, CSS, JavaScript) umgesetzt. Eine besondere Rolle spielt die JavaScript-Bibliothek **Three.js**, mit der Elemente wie Kameras, Lichtquellen oder geometrische Körper für die dreidimensionale Visualisierung erzeugt werden können.

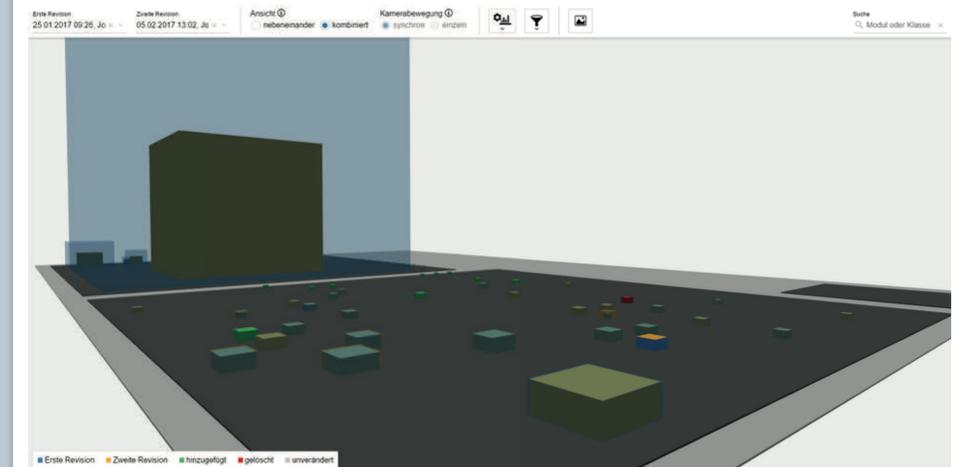


Abbildung 7: Screenshot der Anwendung

Fazit und Ausblick

Basierend auf Konzepten aus der Literatur wurde eine webbasierte Anwendung entwickelt, mit der Struktur und Qualitätsmetriken eines Softwareprojektes visualisiert werden können. Darüber hinaus wurde ein Konzept entwickelt, um zwei verschiedene Versionen des Software-Projektes vergleichen zu können und Unterschiede hervorzuheben.

Für diverse Rollen eines Software-Projektes stellt die Anwendung ein Werkzeug zur Erfüllung verschiedener Ziele dar:

- Architektur des Gesamtsystems nachvollziehen
- mögliche Qualitätsmängel im System aufdecken
- Entwicklung ausgewählter Komponenten beobachten

Durch einen modularen Aufbau wäre die Anwendung mit wenig Aufwand **erweiterbar**, sodass neben Coderadar auch andere Analyse-Werkzeuge die Visualisierung zur Veranschaulichung ihrer Ergebnisse nutzen könnten.

Ideen zur Erweiterung der Benutzeroberfläche bestehen ebenfalls. Beispielsweise könnte die Anwendung konkrete **Handlungsempfehlungen** geben, basierend auf individuell konfigurierbaren **Qualitätsprofilen und kombinierten Metriken**.

Quellenangaben

- [1] L. Erikkh, „Leveraging legacy system dollars for e-business,“ *IT Professional*, Vol. 2, Nr. 3, S. 17–23, Mai/Jun 2000.
- [2] T. Roehm et al., „How do professional developers comprehend software?“ in *34th International Conference on Software Engineering (ICSE)*, Piscataway, NJ, 2012, S. 255–265.
- [3] R. Koschke, „Software Visualization for Reverse Engineering,“ in *Software visualization*, S. Diehl, Hrsg. Berlin Heidelberg: Springer, 2002, Vol. 2269, S. 138–150.
- [4] S. Bassil und R. K. Keller, „Software visualization tools: survey and analysis,“ in *Proceedings / 9th International Workshop on Program Comprehension, IWPC 2001*, Los Alamitos, CA, 2001, S. 7–17.



Konzept und prototypische Implementierung der Modellintegration der Interaction-Room-Methode in die Specification Integration Facility

ZUSAMMENFASSUNG

Der Interaction Room ermöglicht eine effiziente Modellierung von Projektzusammenhängen. Er eignet sich besonders, um ein Verständnis für das Projekt zu erlangen. Die Modellierungssprache zielt dabei auf einen situationsbedingten Detaillierungsgrad ab, auf dem alle Projektbeteiligten kommunizieren können. Doch der Interaction Room ersetzt keine detaillierte Spezifikation, sodass weitere Modellierungssprachen benötigt werden.

Durch die Specification Integration Facility sind die Ergebnisse des Interaction Rooms für andere Modellierungssprachen nutzbar, doch dazu muss eine Modellintegration durchgeführt werden. Aufgrund dessen verfolgt diese Arbeit das Ziel, die Modellintegration durchzuführen und als Konzept darzustellen.

Das Ergebnis der Arbeit ist ein Konzept der Modellintegration, mit dessen Anwendung die Informationen im Interaction Room erschließbar werden. Des Weiteren ist die Modellintegration als Web Service implementiert, um Projekt-Daten vom Interaction Room vollautomatisiert nach der Modellintegration zu transformieren.

INTERACTION ROOM (IR)

Um wesentliche Aspekte eines Systems in Modellen zu erfassen, müssen Sachverhalte auf niedrigem Detaillierungsgrad dargestellt werden. Die Reduktion von Details erfordert jedoch Kenntnisse in der Anwendungsdomäne, um zwischen relevanten und irrelevanten Details unterscheiden zu können. An diesem Punkt setzt die Methode des Interaction Rooms an.

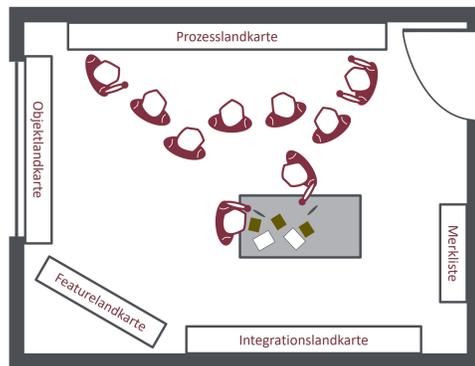


Abbildung 1: Idealtypischer Interaction Room

Der IR hilft allen Projektbeteiligten, ein gemeinsames Verständnis für das zu entwickelnde oder zu verändernde System zu erlangen. An den Wänden des IR werden sowohl fachliche und technische Prozesse und Objekte modelliert, als auch die Systemlandschaft und Interaktionen dargestellt.

SPECIFICATION INTEGRATION FACILITY (SPECIF)

Die SpecIF ist eine Initiative, um Modelle verschiedener Modellierungssprachen in einen gemeinsamen Kontext zu stellen. Eine Zusammenführung von methodenübergreifenden Software-Engineering-Modellen erfolgt durch Abstraktion auf eine Modellierungssprache, die durch fundamentale Elemente Syntax und Semantik verschiedener Modelle nutzbar macht.

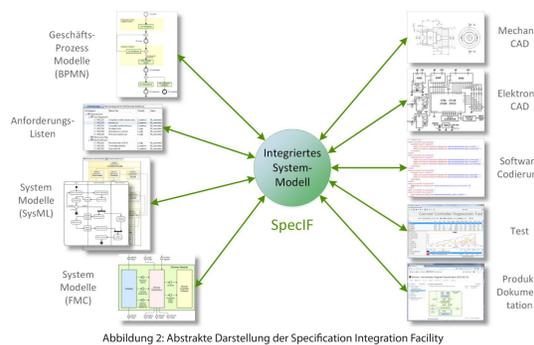


Abbildung 2: Abstrakte Darstellung der Specification Integration Facility

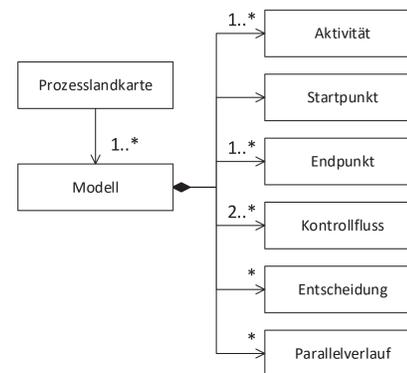
KONZEPT DER MODELLINTEGRATION

Eine Modellintegration beinhaltet die notwendigen Schritte, um Modelle der Interaction-Room-Landkarten in einen gemeinsamen Kontext bringen zu können.

- Darstellungen und Modellelemente trennen
- Modellelemente abstrahieren
- Interrelationen von Modellelementen definieren

Darstellungen und Modellelemente trennen

Um zwischen den Landkarten und den Modellelementen zu unterscheiden, wurden die Metamodelle aufgestellt.



Insgesamt wurden 8 Landkarten und zwei spezielle IR-Elemente untersucht. Das Ergebnis sind 23 identifizierte Elemente der Modellierungssprache.

Modellelemente abstrahieren

Anhand der Metamodelle werden die Modellelemente auf ein erweitertes Vokabular nach „Fundamental Modeling Concepts“ abstrahiert:

- Diagramm
- Akteur
- ◆ Ereignis
- Zustand
- ⚡ Merkmal
- ★ Anforderung

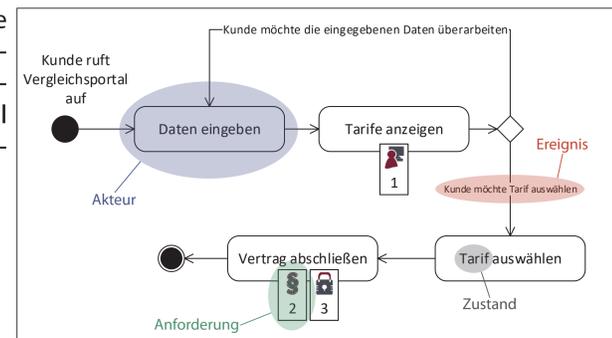


Abbildung 4: Abstraktion eines Modells der IR-Prozesslandkarte

Interrelationen von Modellelementen definieren

Im nächsten Schritt müssen die Modellelemente semantisch verknüpft werden, um die ursprüngliche Bedeutung abzubilden.

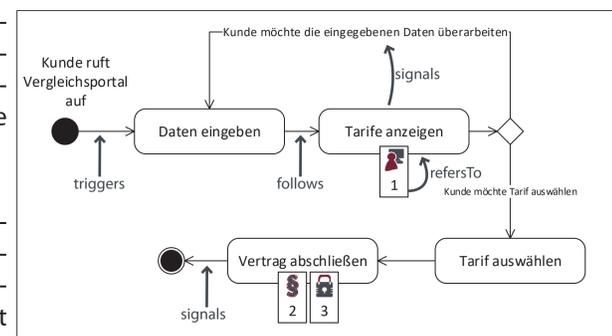


Abbildung 5: Abstraktion der Relationen zwischen Modellelementen

Insgesamt sind 18 verschiedene Relationen zwischen den Elementen definiert. Das Ergebnis ist ein umfassendes semantisches Netz.

PROTOTYPISCHE IMPLEMENTIERUNG

Die prototypische Implementierung umfasst einen Web Service, der unter Berücksichtigung des Architekturmusters **REST** entworfen ist. Das Ziel der Implementierung ist es, eine vollautomatisierte Modellintegration zu erhalten.

| Endpunkt | HTTP-Anfragemethoden |
|--------------|----------------------|
| /project/:id | GET |
| /project | POST |
| /project/:id | DELETE |
| /image/:id | POST |

Der Web Service nimmt IR-Projekte entgegen und erstellt daraus eine Datenstruktur gemäß SpecIF. Gleichzeitig kann der Web Service SpecIF-Projekte anfragen und in ein IR-Projekt transformieren. SpecIF ist zum Zeitpunkt der Entwicklung auf einem ReqIF-Server abgebildet, der für die Persistierung der Daten zuständig ist.



Abbildung 6: Systemlandschaft und Kommunikationsaustausch des Web Services

Die Node.js-Applikation ermöglicht eine betriebssystem-unabhängige Nutzung. Das API ist über das Modul *restify* entwickelt und besitzt einen Zugriffsschutz. Das Austauschformat ist *JSON*, wobei zwei Schemata zur Überprüfung der Nachrichten erstellt wurden.

NUTZEN UND AUSBLICK

Das Konzept der Modellintegration trägt dazu bei, die Verknüpfung von Modellierungssprachen zu formalisieren und voranzutreiben. So wurde bereits das Ergebnis in einem realen Projekt nach dem erstellen Konzept auf SpecIF abgebildet. Daraus können sich folgende Vorteile ergeben:

- Modellierungssprachen in einen gemeinsamen Kontext stellen
- IR-Ergebnisse im Entwicklungsprozess schrittweise verfeinern
- Verlustfreier Austausch von Spezifikationen zwischen Werkzeugen und Organisationen
- Übergang von dokument- zu artefakt-basierter Arbeitsweise

Die Bachelorarbeit wurde in Kooperation mit der Interaction Room GmbH und der Initiative SpecIF entwickelt.

Betreuer: Prof. Dr. Thomas Richter und Simon Grapenthin
Unterstützung seitens SpecIF durch: Dr.-Ing. Oskar von Dungen



Entwicklung eines Redaktionssystems zur Unterstützung von Vertriebsmitarbeitern

Motivation

Das **umfangreiche Portfolio** der adesso AG führt zu Problemen für Vertriebsmitarbeiter im Gespräch mit Kunden:

- Wie können sie einen Überblick über das Portfolio geben?
- Wie können sie bestimmte Kompetenzen hervorheben?
- Wie können unbekannte Chancen gefunden werden?

Um diese Fragen zu beantworten werden zentrale **Marketingressourcen** erstellt, die das Portfolio der Firma beschreiben und darstellen, z.B. Broschüren.

Die adesso AG entwickelt auch ein eigenes Softwaresystem, das diese Aufgabe übernimmt (adesso Portfolio Präsentation, aPP). Dieses besteht bisher aus einer **mobilen Anwendung** mit fest eingetragenen Inhalten.

Ziel dieser Arbeit ist es, **Probleme und Potentiale bei der Inhaltsverwaltung** der Anwendung zu identifizieren und eine entsprechende **Lösung zu entwerfen und zu entwickeln**.

Content Management Systeme

Ein Content Management System (CMS) ist ein Softwaresystem zur **Verwaltung und Publikation von redaktionellen Inhalten**. Mögliche Inhalte umfassen unter anderem Webseiten, Print- oder Online-Medien und technische Dokumentationen.

Der Content Management Prozess beschreibt eine **idealisierte Abfolge des Lebenszyklus** dieser Inhalte.

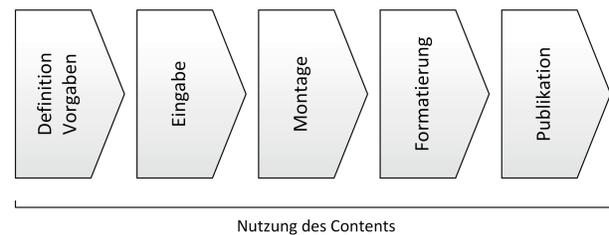


Abbildung 1: Content-Management-Prozess, eigene Abbildung, basierend auf [1, S. 106]

Die Anwendung setzt hierbei lediglich die Schritte 2 und 3 um. Diese Komponente eines CMS bezeichnet [1] auch als **Eingabe-, bzw. Redaktionssystem**.

Problemanalyse

Die Arbeit baut auf der **Clientanwendung** (aPP Client) auf. Sie basiert auf der Universal Windows Plattform und speichert Inhalte in einer lokalen SQLite Datenbank.

Die Inhalte werden **mit der Installation gebündelt** ausgeliefert und in die Datenbank übertragen. Das dafür verwendete Dateiformat ist eine ZIP-Datei, die alle Daten in ein JSON-Dokument und die zugehörigen Bilddateien enthält.

Um die Inhalte zu aktualisieren muss diese Datei **manuell bearbeitet** werden und zusammen **mit der gesamten Anwendung neu installiert** werden. Bearbeiten und Ansehen der Daten ist aufwändig und an Entwickler gebunden. Eine Aktualisierung muss manuell verteilt werden.

Entwicklungsziele

- Editor Oberfläche
- Vorschau
- Freigabeprozess
- Aktualisierungsschnittstelle
- Automatische Aktualisierung im Client
- Authentifizierung und Autorisierung der Nutzer
- Anpassbarkeit

Entwurf

Der Editor (aPP Online) wurde als Webanwendung umgesetzt. Es wurde das neue Framework ASP.NET Core verwendet, um Erfahrungen für dieses zu sammeln.

Die Anwendung wurde auf der Azure Cloud gehostet.

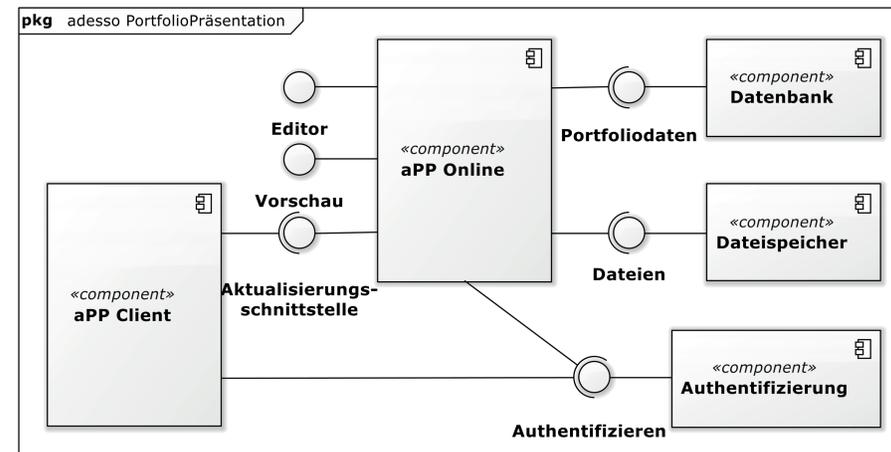


Abbildung 2: UML-Komponentendiagramm des Gesamtsystems

Beispiel Aktualisierungsabfrage

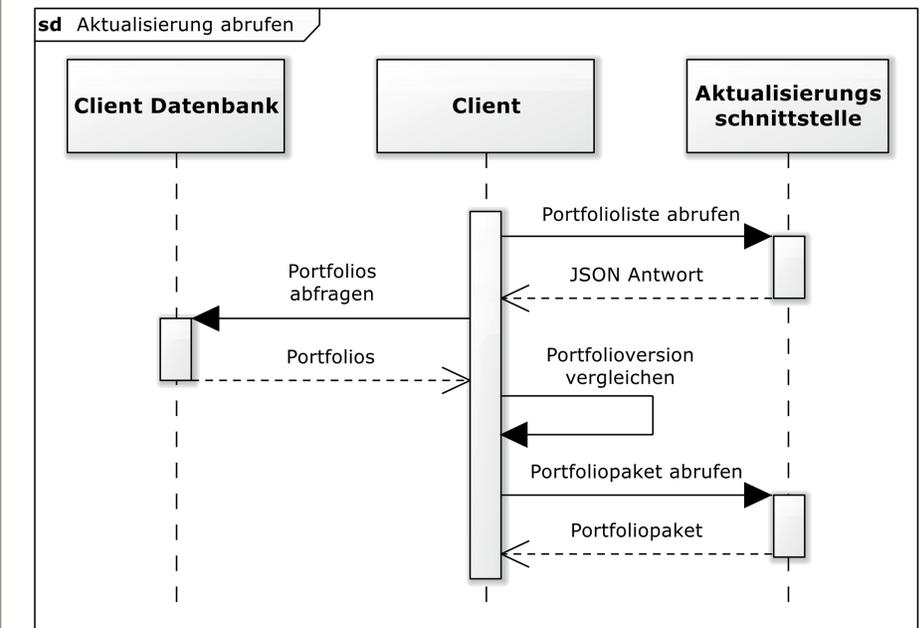


Abbildung 3: UML-Sequenzdiagramm der Aktualisierungsabfrage

Fazit und Ausblick

Aufbauend auf der Problemanalyse wurde ein Redaktionssystem zur Verwaltung des Unternehmensportfolios entwickelt. Es enthält ein Vorschau der Inhalte, die durch eine Freigabefunktion automatisch an aPP Client ausgeliefert werden. Die Anwendung unterstützt **mehrere Portfolios**, sodass verschiedene Abteilungen der adesso AG das System mit eigenen Inhalten nutzen können.

Die Redakteure können eigenständig und zentral die Inhalte aktualisieren, die von Vertriebsmitarbeitern mithilfe von aPP Client den Kunden präsentiert werden.

Die Umsetzung mit ASP.NET Core war einfach und schnell. Die Technologie ist jedoch immer noch jung, wodurch Probleme auftraten:

- **Konflikte** bei der Konfiguration der Authentifizierung
- **Eingeschränkte Funktionalität** beim Datenbank-Mapping mit Entity Framework

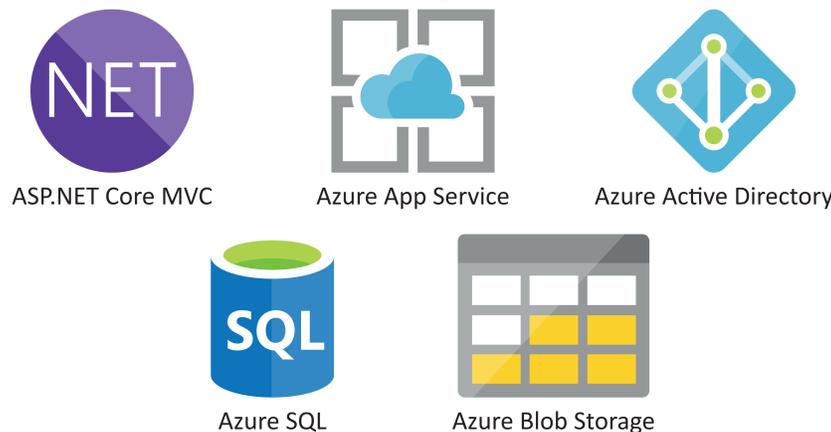
Mögliche Erweiterungen der Anwendung:

- Unterstützung eines **Markup-Editors** für Formatierung in Inhaltsfeldern
- Änderung des Austauschformates: **Layout bereits in aPP Online generieren** und direkt im mobilen Client anzeigen, anstatt es zweifach zu implementieren.

Quellen

[1] L. H. Ehlers, Content Management Anwendungen, Berlin: Logos Verlag, 2003.

Verwendete Technologien



Micro-Frontends: Vergleich von unterschiedlichen Umsetzungsmöglichkeiten im Frontend

Einleitung

Monolithische Web-Applikationen teilen sich eine Codebasis in einer gemeinsamen Laufzeitumgebung. Dabei ist die Anwendung in sich geschlossen und für alle Aufgaben verantwortlich. Webanwendungen mit einer großen monolithischen Codebasis können mit zunehmender Komplexität vor Produktivitäts-, Wachstums- und Innovationsherausforderungen stehen. Micro-Frontends erweitern den Microservice-Gedanken auf das Frontend aus und teilen die Teams in Geschäftsfelder auf, um Funktionen autonom innerhalb eines Teams zu entwickeln.

Microservices gegenüber Micro-Frontends

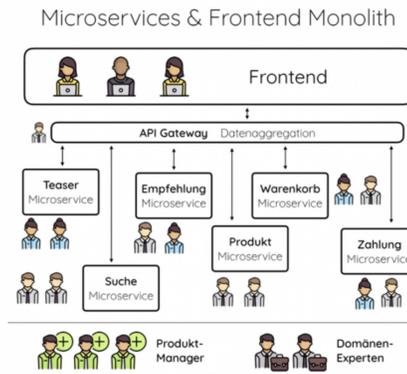


Abbildung 1: Microservices teilen das Backend in verschiedene Services ein, werden aber häufig von einem monolithischen Frontend ergänzt. Dadurch entsteht Abstimmungsaufwand zwischen den Teams.

Micro Frontends

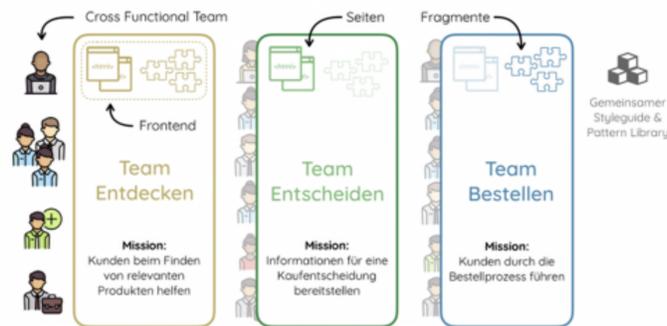


Abbildung 2: Im Micro-Frontends-Modell werden Teams anhand von Kundenbedürfnissen aufgestellt, anstatt an technischen Aspekten. Jedes Team entwickelt autonome Funktionen.

Micro-Frontends Definitionen

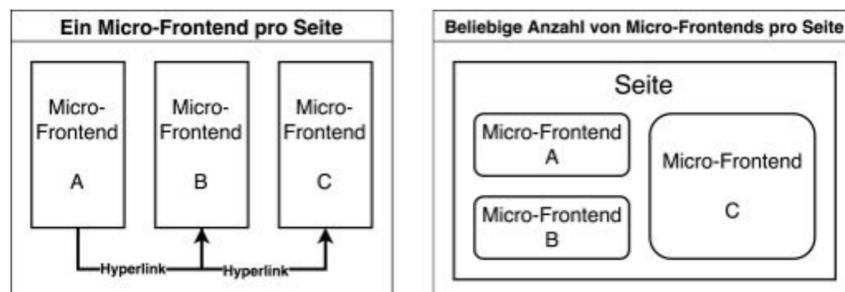


Abbildung 3: Hyperlinks-Integration von Micro-Frontends als Seiten gegenüber von Micro-Frontends als Teil von einer Seite.

Umsetzungsmöglichkeiten

Jeder Seitentyp liegt in der Verantwortlichkeit von genau einem Team. Informationen und Interaktionen aus anderen Vertikalen können über Fragmente (Mini-Applikationen) eingebunden werden.

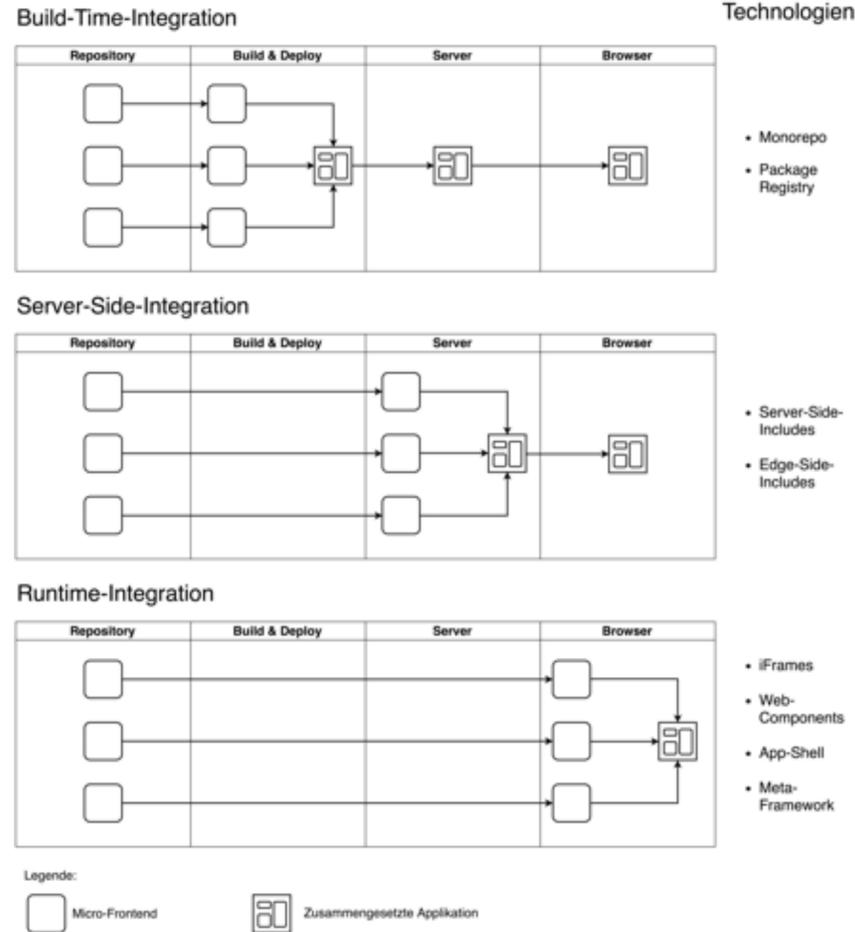


Abbildung 4: Build-Time-, Server-Side- und Runtime-Integration als Umsetzungsmöglichkeit für Micro-Frontends als Teil einer Seite.

Fallstudie

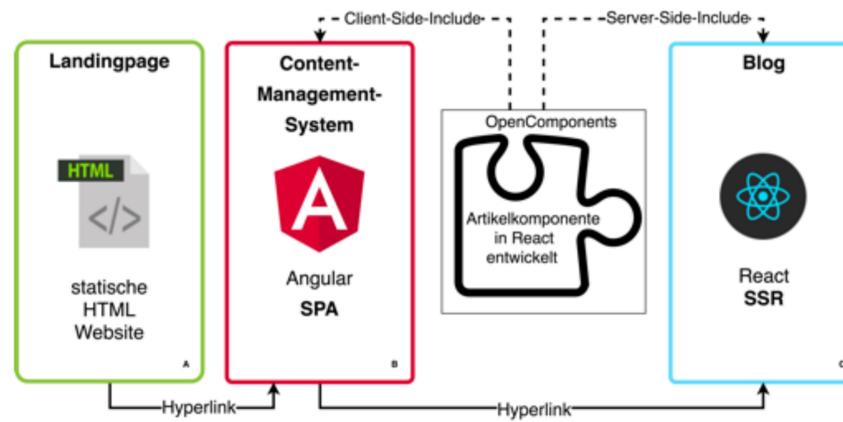


Abbildung 5: Fallstudie, die verschiedene Umsetzungsmöglichkeiten innerhalb eines Projektes einsetzt.

Vergleich

Unterschiedliche Ansätze im Vergleich

| Ansätze | Parallele Entwicklung | Individuelle Deployments | technologische Unabhängigkeit | Robustheit | geringer Kommunikationsaufwand | einheitliche UI | Routing | Isolation | Ressourcenwiederverwendung | Singlizität |
|------------------|-----------------------|--------------------------|-------------------------------|------------|--------------------------------|-----------------|---------|-----------|----------------------------|-------------|
| Hyperlinks | ● | ● | ● | ● | ● | ○ | ○ | ● | ○ | ● |
| Monorepo | ● | ○ | ○ | ○ | ○ | ● | ● | ○ | ● | ○ |
| Package Registry | ● | ○ | ○ | ○ | ○ | ● | ● | ○ | ● | ○ |
| SSR/ESI | ● | ● | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ |
| iFrames | ● | ● | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ |
| Web-Components | ● | ● | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ |
| App-Shell | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Meta-Framework | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

Legende: ● optimal, ○ gut geeignet, ○ geeignet, ○ bedingt geeignet, ○ ungeeignet

Abbildung 6: Vergleichsmatrix der Umsetzungsmöglichkeiten.

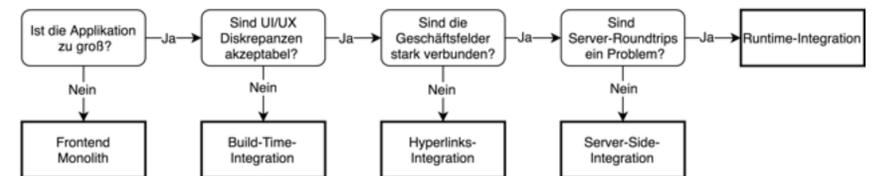


Abbildung 7: Entscheidungsdiagramm für die Umsetzung von Micro-Frontends.

Fazit

Micro-Frontends haben unterschiedliche Definitionen und Anwendungsfälle, sowie mehrere Umsetzungsmöglichkeiten, die alle mit ihren Vor- und Nachteilen kommen. So kann bei der Implementierung von Micro-Frontends mehr Wert auf die Eigenständigkeit der Teams gelegt werden oder auf die strategische Zusammenarbeit. Dabei sollte so viel Teamunabhängigkeit wie möglich und so viel Zentralisierung wie nötig vorgegeben werden. Die Fallstudie zeigt, dass sich mehrere unterschiedliche Umsetzungsmöglichkeiten innerhalb eines Projektes einsetzen lassen. Allgemein sind Micro-Frontends keine Patentlösung für alle Projekte. Micro-Frontends können die Probleme von Frontend-Monolithen lösen, allerdings entstehen durch verteilte Systeme neue Herausforderungen. Deswegen ist es wichtig die verschiedenen Möglichkeiten und Herausforderungen von Micro-Frontends zu kennen.

Hintergrund

Im Rahmen einer Bachelor-Thesis wurde die Micro-Frontends-Architektur, sowie dessen Umsetzungsmöglichkeiten von Denis Paris untersucht. Betreut wurde die Arbeit dabei durch Prof. Dr. Thomas Richter an der Hochschule Rhein-Waal Kamp-Lintfort und Herrn Alexander Henka M.Sc. von der adesso AG.