

# Smart Enterprise Project Cockpit

## Bachelorarbeit

Abteilung Informatik  
Hochschule für Technik Rapperswil

Frühlingssemester 2019

**Autoren:** Lukas Steiger  
Pascal Hürlimann

**Betreuer:** Dr. Daniel Keller, HSR

**Experte:** Prof. em. Hans Rudin

**Gegenleser:** Prof. Dr. Peter Heinzmann, HSR

**Projektpartner:** Dr. Thomas Weh, Siemens Schweiz AG

# Inhaltsverzeichnis

Dokumentklassifikation .....	1
Abstract .....	2
Management Summary .....	3
1. Einführung .....	7
1.1. Ausgangslage .....	7
1.2. Rückblick auf die Studienarbeit .....	7
1.3. Über VCR .....	10
2. Projektmanagement .....	15
2.1. Projektvision & Kontextabgrenzung .....	18
2.2. Rahmenbedingungen, Artefakte & Lieferergebnisse .....	21
2.3. Meilensteine .....	22
3. Anforderungsanalyse .....	24
3.1. Einführung .....	24
3.2. Meilensteinbewilligungsprozess .....	26
3.3. Nichtfunktionale Anforderungen .....	36
4. Software Architektur .....	38
4.1. Rahmenbedingungen .....	38
4.2. Lösungsstrategie .....	39
4.3. Bausteinsicht .....	44
4.4. Laufzeitsicht .....	58
4.5. Verteilungssicht .....	66
4.6. Querschnittliche Konzepte .....	70
4.7. Risiken .....	75
5. Ergebnisse & Ausblick .....	81
5.1. Umgesetzte Funktionalität .....	82
5.2. Zielerreichung & Umsetzungsgrad .....	93
5.3. Ausblick .....	97
5.4. Schlussfolgerungen .....	99
6. Glossar .....	100
7. Quellenverzeichnis .....	106
8. Anhang .....	110
8.1. Persönliches Fazit .....	110
8.2. Metriken .....	112
8.3. Prototypen & Wireframes .....	114

8.4. Evaluationen .....	117
8.5. Einsatz von Third Party Software .....	121
8.6. SQL Code Ausschnitt .....	129
9. Auszüge aus der Studienarbeit .....	130
9.1. Ziele & Stakeholder .....	131
9.2. Nutzungsanalyse .....	133
9.3. Bestandsanalyse .....	137
9.4. Personas .....	143
9.5. Benutzungsszenarien .....	148

## Dokumentklassifikation

Diese Ausgabe der Bachelorarbeit ist freigegeben für die öffentliche akademische Publikation auf <https://eprints.hsr.ch/>. Um diese öffentliche Publikation überhaupt zu ermöglichen, wurden einige Kapitel oder Abschnitte der eigentlichen Bachelorarbeit von der Publikation ausgeschlossen. Dies war unumgänglich, da die von der Publikation ausgeschlossenen Teile vertrauliche Informationen enthalten.

Die Auslassung dieser Dokumentteile kann dazu führen, dass der Lesefluss leicht beeinträchtigt ist. Dennoch ziehen wir diese gekürzte Variante vor, da die Alternative eine Unterlassung der Publikation gewesen wäre.

## Abstract

Bei [\[Siemens\]](#) wurde bis vor gut zehn Jahren die Fortschritts- und Kostenverfolgung eines Projektes in jedem Projekt individuell gehandhabt. Es gab wenig einheitliche Vorgaben und keinerlei Unterstützung durch geeignete Software.

Erste Pioniere haben 2005 damit begonnen ein Werkzeug zu entwickeln - **VCR** - welches Projektbeteiligte und Führungskräfte bei projektbezogenen Aufgaben unterstützt, auch projekt- und standortübergreifend. Der Funktionsumfang von VCR lässt sich gut mit seinem vollen Namen beschreiben: *Visualization, Controlling* und *Reporting*.

VCR steht bei [\[Siemens\]](#) heute klar im Mittelpunkt, wenn es um projektbezogene Daten geht. So wurden bisher über 1700 Projekte mit Meilensteinen in VCR abgewickelt, wobei sich die einzelnen Projekte in Bezug auf Projektgrösse, Dauer, Umfang und verwendeten Geschäftsprozess stark unterscheiden.

Aktuell entsprechen *Architektur, Benutzbarkeit, Performanz* und *Technologie* von VCR nicht mehr den heutigen Anforderungen. An gewissen Ecken gibt es ungenutzte Funktionen, wiederum werden gewisse Abläufe ausserhalb durchgeführt, obwohl sie ins VCR gehören würden. Eine grundlegende Überarbeitung der Architektur ist zudem vor einigen Jahren gescheitert.

Wir haben uns grundsätzlich mit VCR und seinen Umsystemen auseinandergesetzt, Handlungsfelder aufgezeigt und eine Strategie entwickelt, wie man VCR Schritt für Schritt durch eine neue Lösung ersetzen kann. Wir haben mit dem Meilensteinbewilligungsprozess einen ersten Workflow aus dem VCR Ökosystem in einer neuen Lösung umgesetzt. Die neue Lösung ist mit VCR verbunden und bezieht sämtliche existierenden Daten von dort. Die von uns gewählte Architektur und Technologien erlauben es, VCR Schritt für Schritt mittelfristig abzulösen. Durch diesen iterativen Prozess kann sichergestellt werden, dass eine neue Lösung sämtliche neuen Bedürfnisse abdecken kann und gleichzeitig die gewohnte Kontinuität von VCR weiterhin gegeben ist.

# Management Summary

## Ausgangslage

In Zusammenarbeit mit unserem Projektpartner [\[Siemens\]](#) haben wir im Rahmen der Studienarbeit "NextGen Project Management Tooling" die bestehenden Projektmanagement-abläufe im Umfeld der Fortschritts- und Kostenverfolgung bei Siemens analysiert. Schnell wurde klar, dass die Applikation VCR dabei die zentrale Rolle einnimmt.

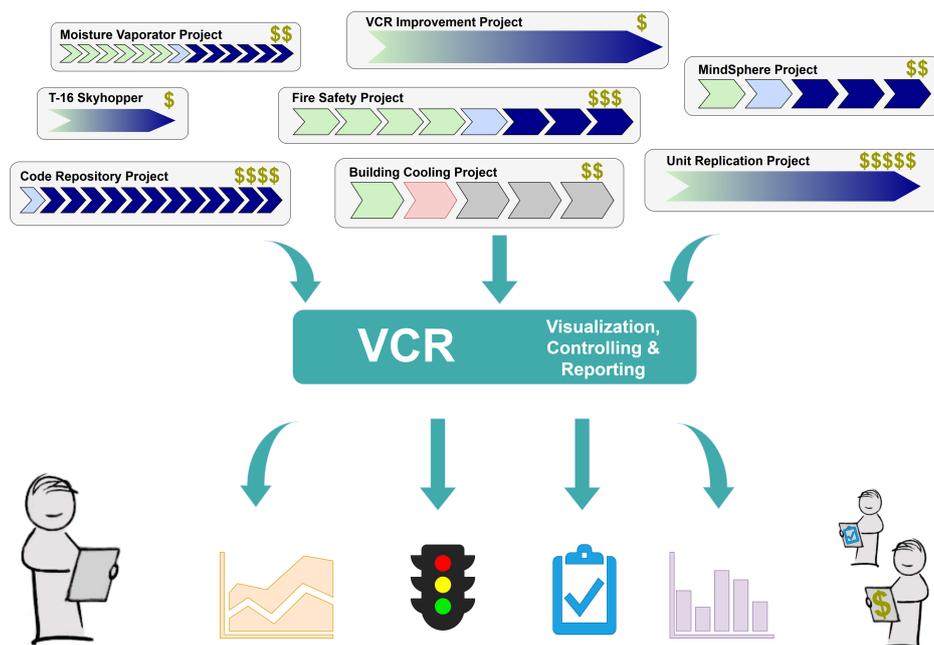


Abbildung 1. Bei Siemens spielt heute die Software VCR eine zentrale Rolle, wenn es um Projektfortschritt, Kostennachverfolgung und Reporting geht, wobei verschiedene Prozesstypen unterstützt werden.

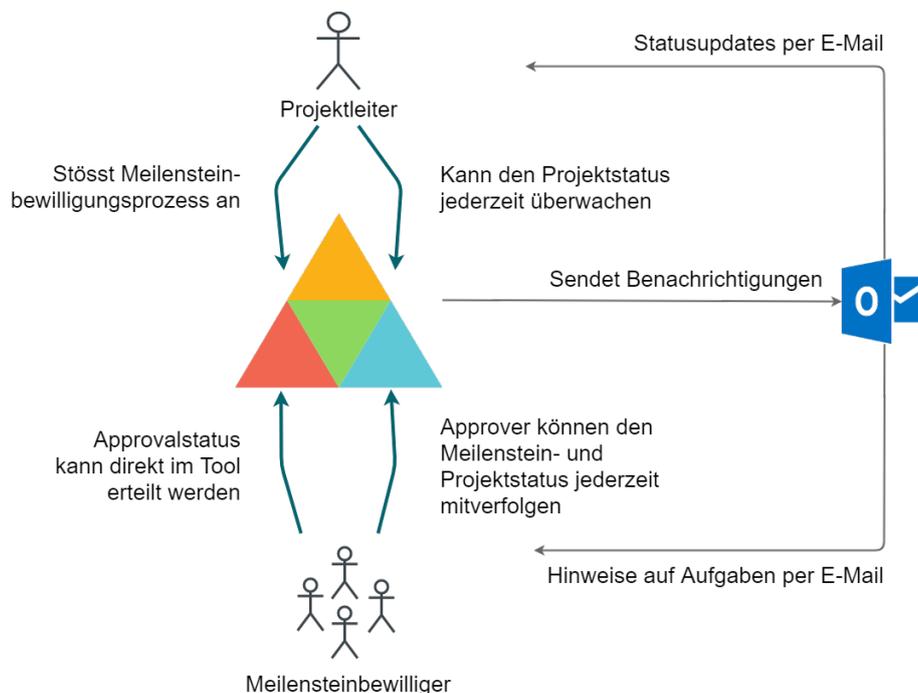
VCR selbst, aber auch viele Prozesse darum herum sind während 15 Jahren historisch gewachsen, wurden jedoch nie mehr grundsätzlich überarbeitet. Nach der Analyse im ersten Schritt wurde durch uns in einem zweiten Schritt ein Prototyp entwickelt, welcher einige der Ungereimtheiten im heutigen Ökosystem adressiert. Allerdings war dieser Prototyp komplett losgelöst von VCR. Nach Beendigung der Studienarbeit wurde zusammen mit Siemens eine Vision definiert: **VCR soll Schritt für Schritt durch eine zeitgemässe Neuentwicklung abgelöst werden.** Eine Neuentwicklung wird aufgrund des beträchtlichen Funktionsumfangs von VCR einige Jahre in Anspruch nehmen, weswegen wir uns im Rahmen der Bachelorarbeit auf einzelne Teilaspekte beschränken müssen. So soll geprüft werden, ob ein Parallelbetrieb von VCR und unserer neuen Lösung während dieser mehrjährigen Übergangsphase möglich ist und wie ein solcher strategisch umgesetzt werden kann. Im Rahmen dieser Bachelorarbeit soll der Prototyp aus der Studienarbeit in eine operative Lösung überführt und erste ausgewählte Funktionalität in einer neuen Lösung umgesetzt werden.

## Vorgehen

Als erste neu umzusetzende Funktionalität wurde gemeinsam mit dem Auftraggeber der bereits während der Studienarbeit identifizierte **Meilensteinbewilligungsprozess** ausgewählt. Um eine Ablösung dieses Prozesses zu entwickeln, haben wir uns zuerst vertieft mit dem existierenden Prozess befasst. Dazu haben wir sowohl die organisatorischen Aspekte, aber auch die technische Umsetzung eingehend analysiert, um ein umfassendes Verständnis aller involvierten Workflows zu erlangen.

In einem nächsten Schritt haben wir eine Strategie entwickelt, wie der aus der Studienarbeit entstandene Prototyp in die existierende VCR Landschaft integriert werden kann. Dazu musste zusammen mit unserem Auftraggeber definiert werden, wie die Applikation bei Siemens ausgeliefert und integriert wird und wie bestehende Infrastruktur effektiv genutzt werden kann. Ein essentieller Teil der Integrationsstrategie ist der Datenaustausch mit dem bestehenden VCR Ökosystem. Die neue Applikation muss dabei auf die VCR Stammdaten zugreifen, um Datenkonsistenz garantieren zu können.

Wöchentliche Sitzungen mit dem Auftraggeber und dem Betreiber von VCR haben garantiert, dass der Umsetzungsfokus stets den von Siemens gewünschten Aspekten entsprach. Mit diesem iterativen Vorgehen konnten wir sowohl den Fortschritt aufzeigen, gleichzeitig aber zeitnahe Rückmeldungen einholen, um zielgerichtet an der neuen Lösung arbeiten zu können.



*Abbildung 2. Ablaufdiagramm des neuen, vereinfachten Meilensteinbewilligungsprozesses. Der gesamte Ablauf wird von unserer Applikation gesteuert. Die Anzahl der direkt involvierten Applikationen kann von drei auf eine reduziert werden, da Email nur noch als Erinnerungsmedium genutzt wird.*

## Ergebnisse

Basierend auf den Ergebnissen der Studienarbeit wurde eine neue Lösung unter Verwendung eines modernen, aber bewährten Technologiestacks entwickelt. Mehrere Mängel der bisherigen Lösung wurden behoben, insbesondere in Bezug auf Benutzerfreundlichkeit und Performanz. Je nach Rolle können die an einer Meilensteinbewilligung beteiligten Personen nun den Verlauf einer solchen Genehmigung mit einem einzigen Tool überwachen und steuern. Dies ist eine große Verbesserung gegenüber der vorangehenden, teilweise instabilen Lösung, wo VCR, Excel und Email gleichermaßen involviert waren. Die neue, ansprechende Webanwendung kann zur Freude der Endbenutzer auch auf mobilen Geräten genutzt werden und ermöglicht neue Nutzungsszenarien. Diese anwenderorientierten Verbesserungen werden durch einen kontinuierlichen und zuverlässigen Datenaustausch zwischen dem alten und dem neuen System ergänzt. Da die Datenkonsistenz im Umgang mit kritischen Geschäftsdaten von grösster Bedeutung ist, können wir mit diesem Ansatz eine gemeinsame Datenbasis gewährleisten. Das neue Cockpit zur Meilensteinbewilligung ermöglicht eine konsistentere Datenverarbeitung und eine strengere Kontrolle des Projektfortschritts, ohne auf bisher gewohnte Funktionalität verzichten zu müssen.

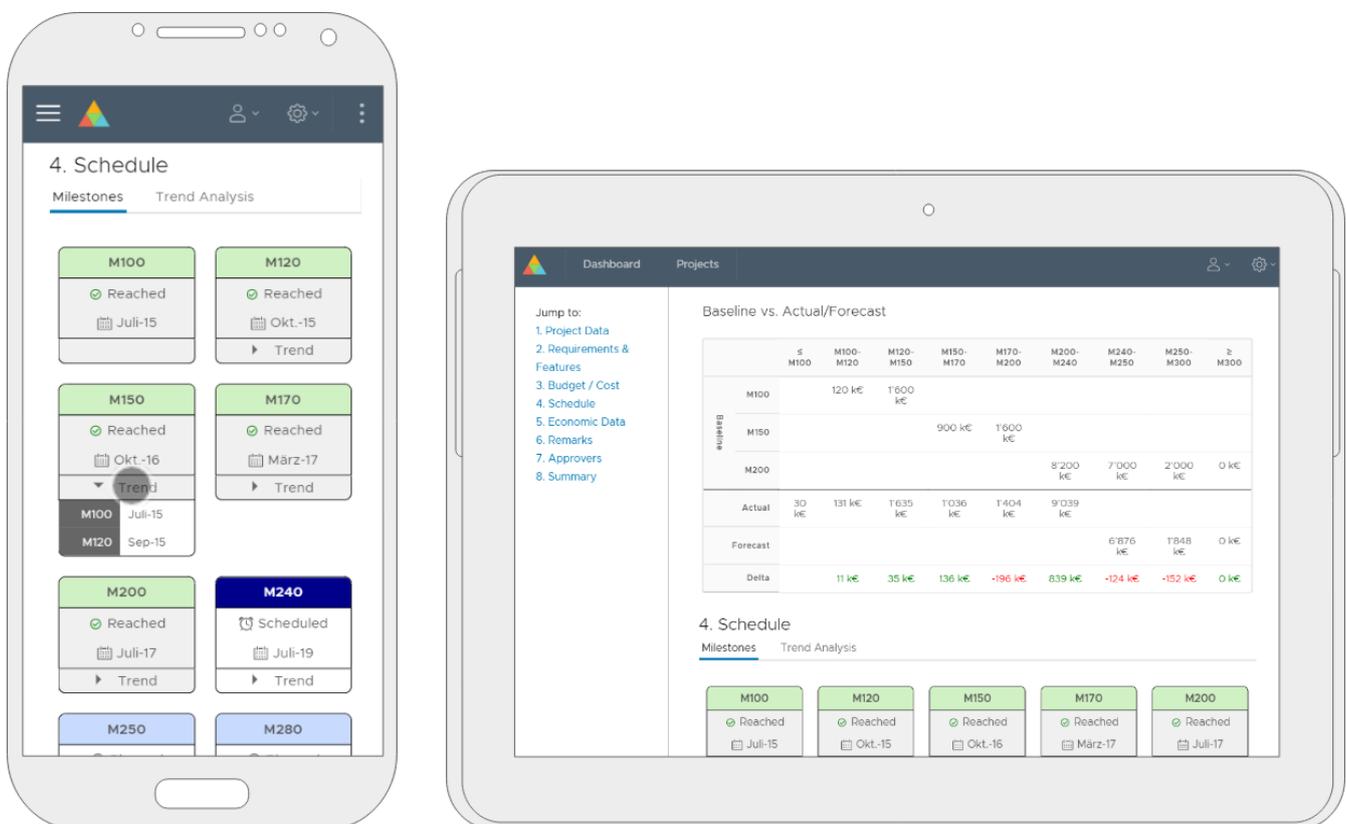


Abbildung 3. Alle für eine Meilensteinbewilligung notwendigen Projektkennzahlen werden gesammelt, aufbereitet und durch unsere neu entwickelte Applikation visualisiert.

## Ausblick

Es hat sich gezeigt, dass es möglich ist, einen bestimmten Workflow auszuwählen, zu isolieren und einen Nachfolger dafür zu entwickeln. Der hier verwendete Problemlösungsansatz ist ein Beispiel dafür, dass die schrittweise Modernisierung einer Anwendung ein praktikabler Ansatz ist, wenn man sicherstellt, dass das Altsystem und die neue Anwendung während der Übergangszeit parallel betrieben werden können.

Nun liegt es an unserem Auftraggeber Siemens, die letzten noch aus der Arbeit verbliebenen offenen Punkte des Meilensteinbewilligungsprozesses umzusetzen. Da die Infrastruktur für die Applikation bereits betriebsbereit ist, kann die Applikation mit wenig Aufwand für Endanwender freigeschaltet werden.

Die Ablösung weiterer Abläufe unter Verwendung desselben Lösungsansatzes kann danach mit dem gleichen Ansatz vorangetrieben werden. So wäre es möglich, VCR innert weniger Jahre durch das neue System abzulösen, indem durch Anwendung einer agilen Vorgehensweise Schritt für Schritt Abläufe und Funktionen von VCR portiert werden.

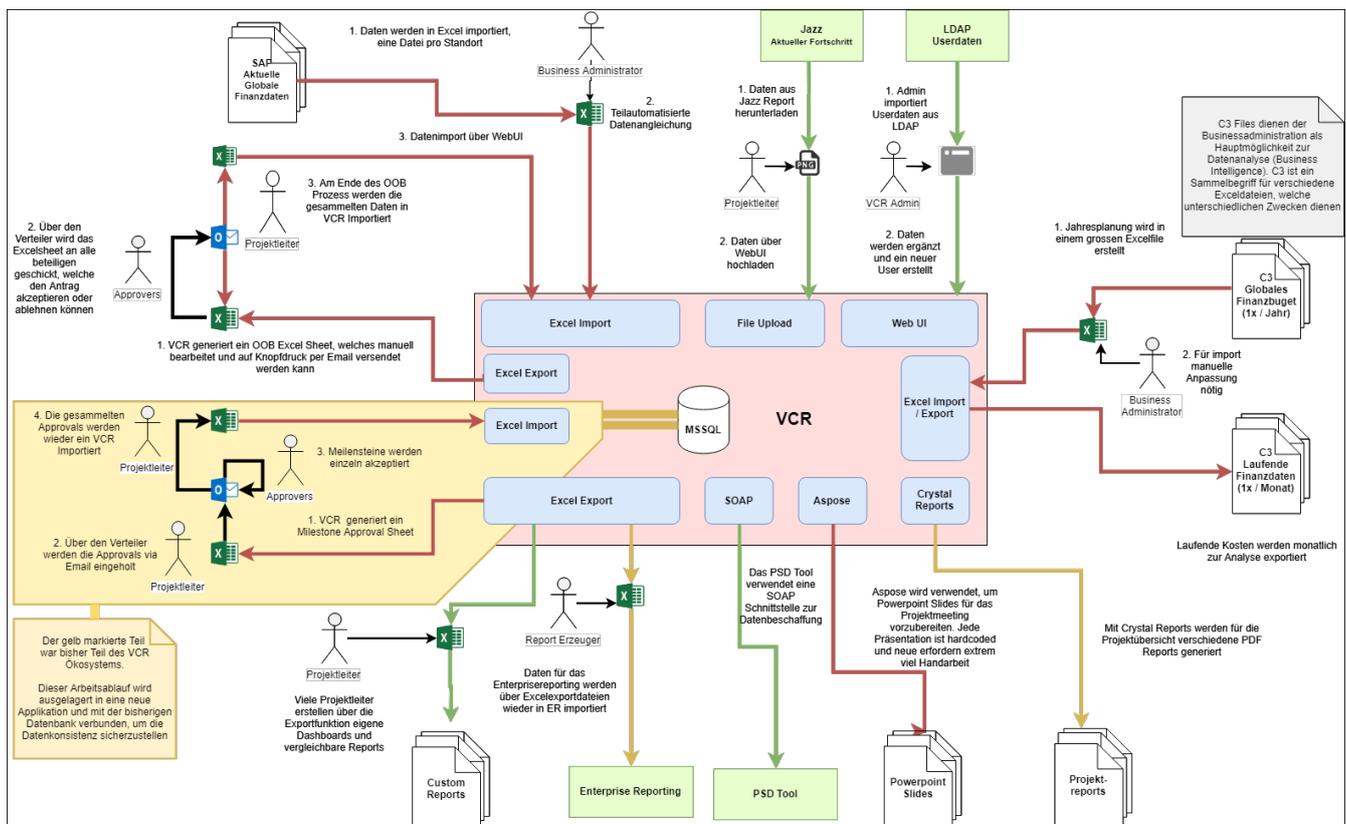


Abbildung 4. Gelb hervorgehoben ist der durch uns abgelöste Meilensteinbewilligungsprozess. Dies ist jedoch nur der Anfang, da noch viele weitere Abläufe abgelöst werden müssen.

# 1. Einführung

## 1.1. Ausgangslage

Unser Auftraggeber bei Siemens verantwortet den Bereiche *Prozesse, Tools* und *Infrastruktur* für die weltweit tätige Business Unit *Building Products*. In diesen Verantwortungsbereich fallen auch die Weiterentwicklung und der Betrieb von verschiedenen Softwarelösungen, welche im Umfeld der Fortschritts- und Kostenverfolgung eingesetzt werden. Das heutige Ökosystem ist in die Jahre gekommen und Bedarf mittelfristig einer zukunftsweisenden Veränderung, um mit der technischen Entwicklung und den neuen technischen Möglichkeiten Stand zu halten. Ein zentraler Aspekt dieses Ökosystems ist die Applikation VCR (Visualization, Controlling und Reporting). Diese Applikation unterstützt Projektbeteiligte und Führungskräfte bei projektbezogenen Aufgaben.

Eine grundlegende Überarbeitung der Architekturprobleme von VCR und seinem Ökosystem ist vor einigen Jahren an den technischen Herausforderungen und der fehlenden strategischen Herangehensweise gescheitert. Um neue Impulse für eine nachhaltige und zukunftsorientierte Weiterentwicklung zu erhalten, hat Siemens die Unterstützung der HSR ersucht, um neue Impulse und Lösungsansätze zu gewinnen.

Mit dem Ziel, eine Vision für eine zukunftsweisende Lösung zu finden und die Vision mit einem Prototyp zu verifizieren, wurde die Studienarbeit "*NextGen Project Management Tooling*" bei der HSR eingereicht und Herr Daniel Keller hat sich dankenswerterweise dazu bereit erklärt, diese Arbeit zu betreuen. Die Aufgabenstellung und das Ergebnis der Studienarbeit sind nachfolgend erläutert, um die Ausgangslage für diese Bachelorarbeit aufzuzeigen.

## 1.2. Rückblick auf die Studienarbeit

Diese Bachelorarbeit basiert auf den Ergebnissen der Studienarbeit [\[SA\]](#) "*NextGen Project Management Tooling*". Dieses Kapitel soll als Heranleitung an diese Bachelorarbeit sowie als Abgrenzung zur bereits abgeschlossenen Studienarbeit dienen.

### 1.2.1. Aufgabenstellung der Studienarbeit

*Die uns im Rahmen der Studienarbeit überreichte Aufgabenstellung liess sich grob in zwei Teilaufgaben unterteilen. Im ersten Schritt ging es darum, die Kundenbedürfnisse zu verstehen, indem die aktuellen Projektmanagement Workflows im Umfeld der Fortschritts- und Kostenverfolgung, wie sie typisch für ein Grossunternehmen sind, analysiert wurden. Mit unserer Außenperspektive konnten wir die bestehende Projektmanagement Toollandschaft unabhängig bewerten, um daraus eine zukunftsweisende Vision für die verschiedenen Anwendungszwecke zu entwerfen. Als zweite integrale Aufgabe der Arbeit wurden wir gebeten, eine Prototyp-Anwendung zu implementieren, die einige Pain Points des aktuellen Ökosystems adressiert, um zu zeigen, wie eine zukünftige Landschaft aussehen könnte und was die Vorteile einer solchen Entwicklung sein würden.*

— [\[SA-1\]](#)

### 1.2.2. Ergebnis der Studienarbeit

Als Teil der eingehenden Analysephase hatten wir unter anderem ein Kontextdiagramm entwickelt, welches die bisherigen Daten- und Informationsflüsse visualisiert. Des Weiteren wurde durch uns ein Datenmodell erarbeitet, welches die Kernbedürfnisse der Systeme abdeckt und als solide Grundlage für den Prototypen diente, welchen wir im zweiten Teil der Studienarbeit entwickelt hatten. Als Ergebnis der Studienarbeit hatten wir ein simples Projektcockpit entwickelt, mit Fokus auf Benutzbarkeit, Erweiterbarkeit und Performanz.

*Relevante Auszüge aus der Studienarbeit sind am Schluss dieser Arbeit im Kapitel [Auszüge aus der Studienarbeit](#) zu finden.*



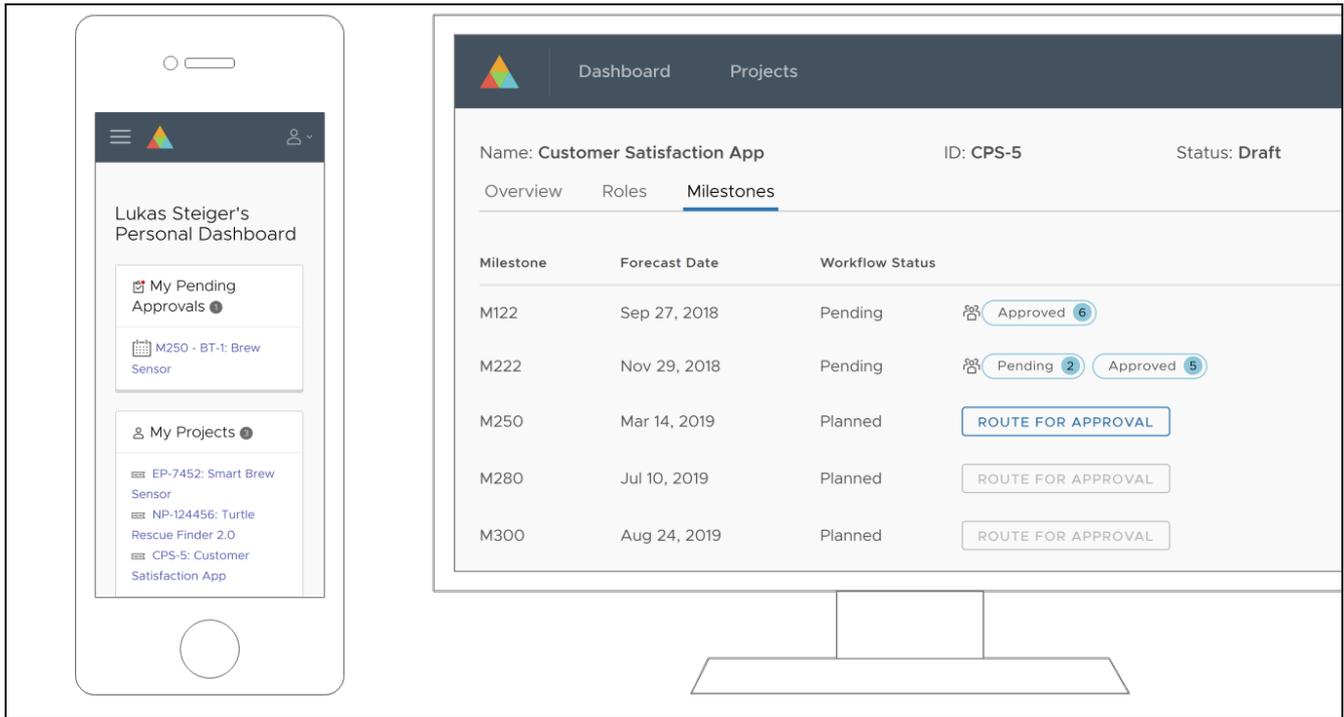


Abbildung 6. Der aus der Studienarbeit [SA] resultierende Prototyp zeigt ein persönliches Dashboard auf einem Mobilgerät und die Meilenstein Übersichtsseite in der Desktopansicht.

### 1.3. Über VCR

Das vorangehend visualisierte Kontextdiagramm zeigt klar auf, dass im Umfeld der Fortschritts- und Kostenverfolgung bei Siemens das Tool VCR (Visualization, Controlling und Reporting) eine zentrale Rolle spielt.

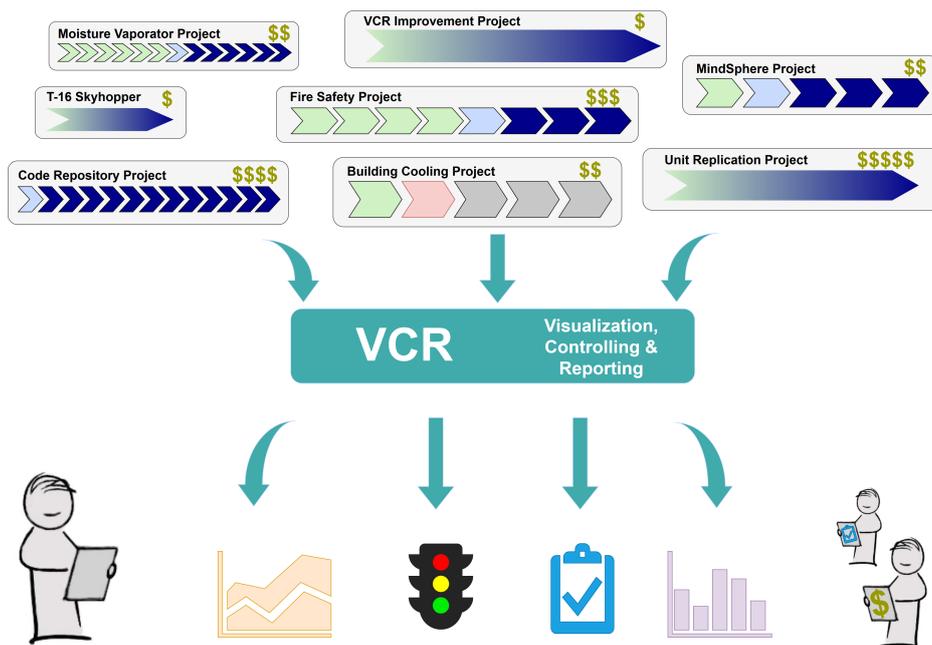


Abbildung 7. Die bestehende Applikation VCR verwaltet sämtliche Projekte bei [Siemens]. Es gibt zahlreiche Schnittstellen und Workflows. Zudem gibt es diverse Reports, welche erstellt werden können.

### 1.3.1. Historische Entstehung

Bis ins Jahr 2005 wurde die Projektfortschrittsnachverfolgung bei *Siemens* in sämtlichen Projekten individuell gehandhabt. Für Führungskräfte oder Stabsstellen wie das Qualitätsmanagement war es äusserst schwierig, sich einen Überblick über Projektgrenzen hinweg zu verschaffen, da es keinerlei Gemeinsamkeiten in der Art der Projektführung gab. Als Werkzeug dominierte klar *Microsoft Excel*, wobei jeder Projektleiter "sein eigenes Ding" gemacht hat ([Schaefer18]).

Dieser Missstand wurde zuerst an einem Siemens Standort in Deutschland erkannt. Dort wurde daraufhin innerhalb einer Teilorganisation eine einfache, auf *Microsoft Access* basierende Lösung geschaffen, um einen ersten Schritt in Richtung Vereinheitlichung zu gehen. Aufgrund der positiven Resonanz auf diese Lösung wurde daraufhin eine zentrale Lösung für [Siemens] erarbeitet, unter Führung von *Guido Schneider* ([Schneider19]).

Unique-PID	Location	Agile	PL	PQM	Main-/Single-/SubProject Name
CP-75646	BOG	False	Lopez, Rosa	Hernandez, Hector	App for Latin American Market
NP-124456	GPS	False	Sanchez, José	Unassigned	Turtle Rescue Finder
EP-7452	MUC	False	Müller, Horst	Kramp, Anne	Smart Brew Sensor
CP-74564	ZUG	True	Steiger, Lukas	Huerlimann, Pascal	Customer Satisfaction App
EP-128	SFO	False	Sanders, Jack	Hunter, Sarah	Bridge Connector
NP-7845315	GPS	False	Unassigned	Unassigned	Turtle Rescue Finder 2.0

Abbildung 8. Screenshot einer Projektübersicht aus der heutigen Lösung VCR.

### 1.3.2. Entwicklung bis Heute

Seit dem Beginn der Entwicklung Mitte der 00er-Jahre ist an VCR stetig weiterentwickelt worden. Zu Beginn wurden viele Features in kurzen Zyklen entwickelt, mit der Zeit wurden die Zyklen jedoch deutlich länger, um den höheren Ansprüchen an die Qualitätssicherung gerecht zu werden ([Schneider19]). Seit mehr als fünf Jahren werden pro Jahr ungefähr zwei bis vier

Feature-Releases veröffentlicht. Aktuell ist die Version 2.21.x von VCR im produktiven Einsatz.

Vor einigen Jahren wurde zudem der Versuch gestartet, in einer Version 3.0 tiefgreifende architektonische Anpassungen vorzunehmen. Dieses Vorhaben ist jedoch gescheitert, da mit der Zeit die parallele Weiterentwicklung (2.x) und die Überarbeitung (3.x) zu stark auseinandergelaufen sind und die Überarbeitung aus Kapazitätsgründen oft monatelang zurückgestellt wurde.

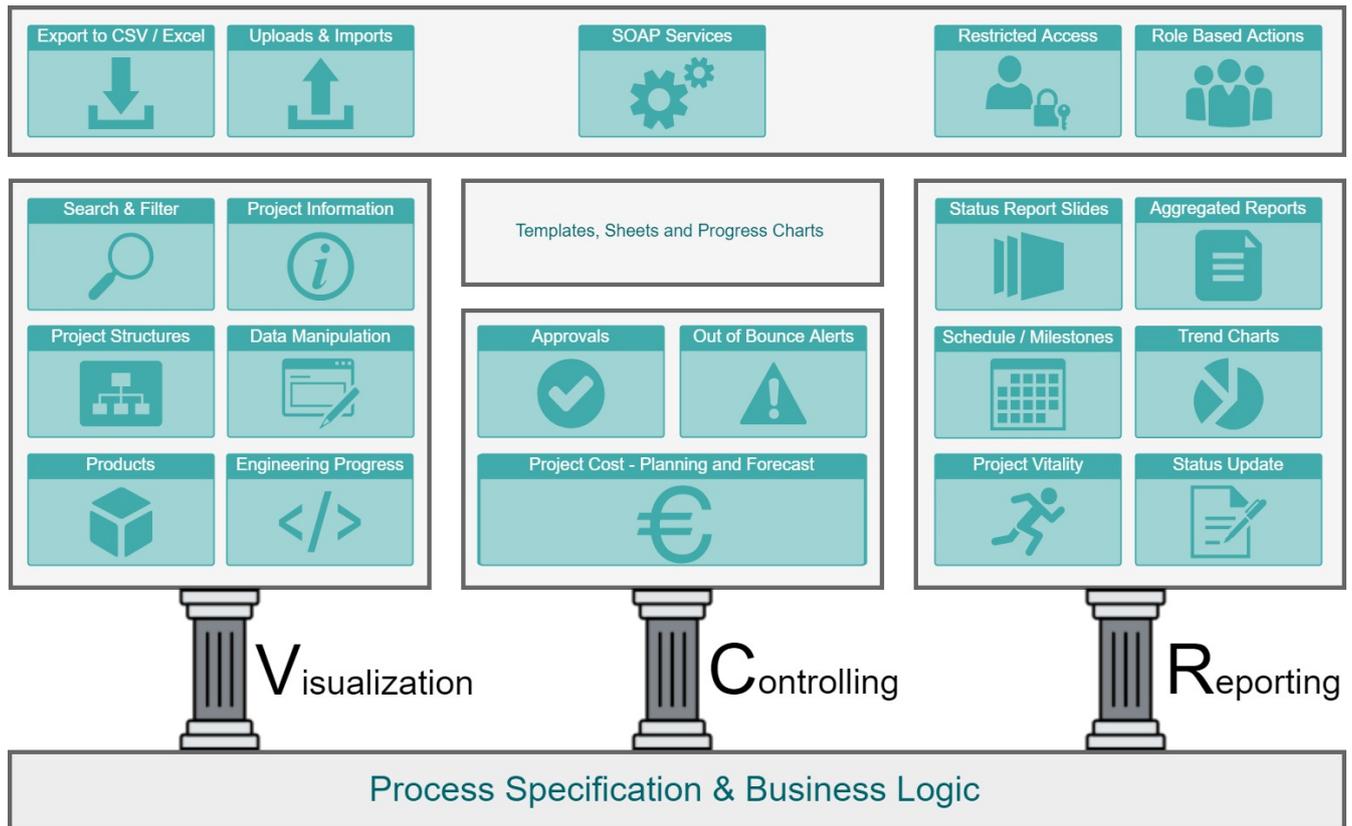


Abbildung 9. Ein Versuch, das heutige VCR anhand seiner drei Grundpfeiler Visualization, Controlling und Reporting verständlich darzustellen.

Es steht fest, dass VCR heute bei [\[Siemens\]](#) eindeutig im Mittelpunkt steht, wenn es um projektbezogene Daten geht. So wurden seit der Einführung von VCR über 1700 Projekte mit Meilensteinen darin abgewickelt, wobei sich die einzelnen Projekte voneinander stark unterscheiden. So ist die Spannweite mit sehr kleinen Projekten mit Gesamtbudget von wenigen 10'000 Euro bis zu Grossprojekten im zweistelligen Millionenbereich alles vorhanden. VCR unterstützt zudem verschiedene Geschäftsprozesse, wobei jedes Projekt auf einem dieser (aktuell sechs) Prozesstypen basieren muss.

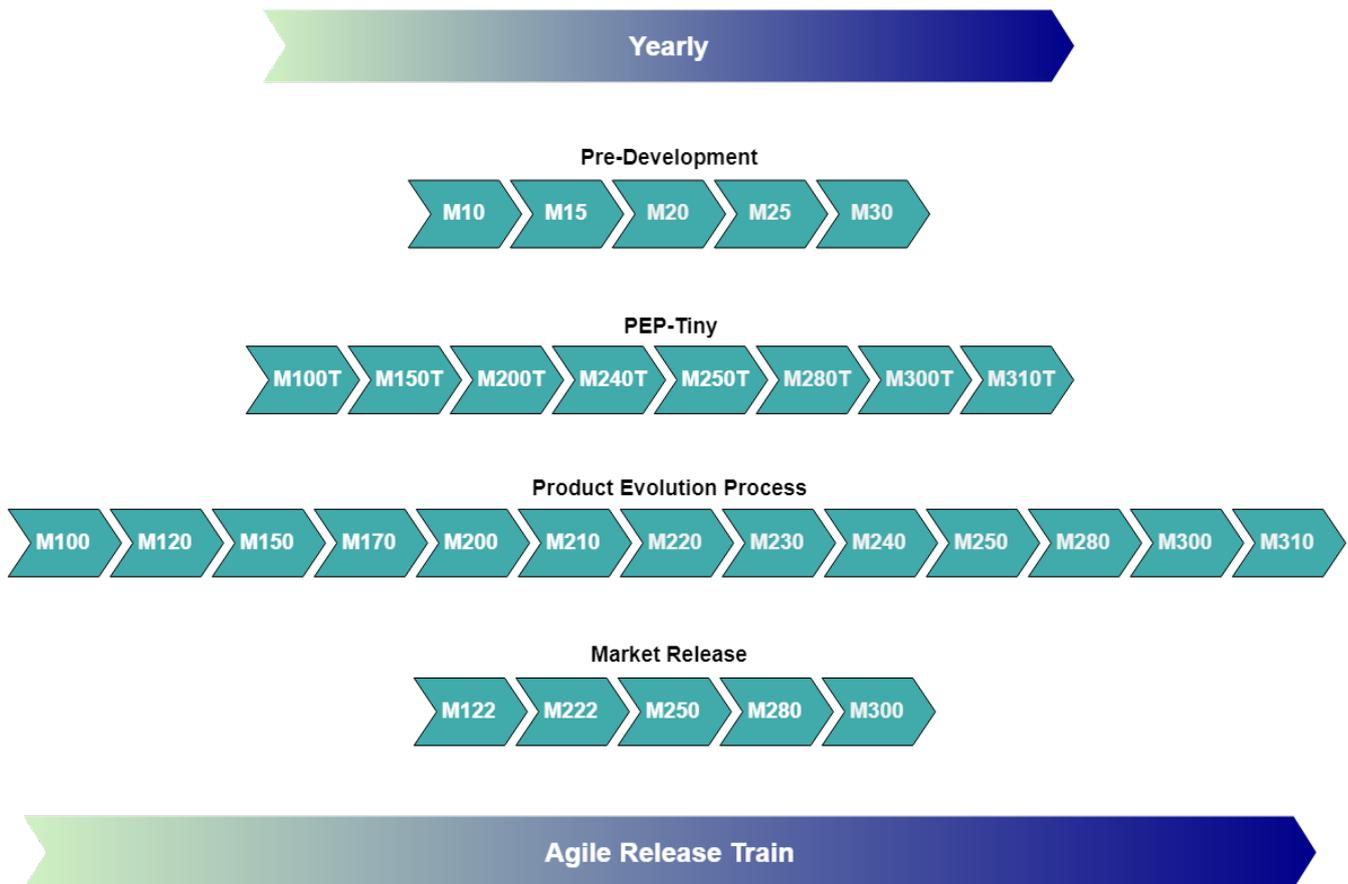


Abbildung 10. Aktuell bei [Siemens] im Einsatz stehende Geschäftsprozessstypen. Die vier mittleren Projekttypen haben eine variierende Anzahl Meilensteine. Die beiden Prozessstypen Yearly und Agile Release Train wiederum kennen keine Meilensteine.

Eine weitere wichtige Komponente von VCR, neben der Möglichkeit zur Bearbeitung von Projektdaten, ist deren Visualisierung und Berichterstattung. VCR bietet eine breite Auswahl von Visualisierungen, welche verschiedene Aspekte der Projektauswertung abdecken. Der Projektfortschritt lässt sich an verschiedenen Metriken festhalten, wie beispielsweise mit einer Meilensteintrendanalyse, oder auch einer Budgetanalyse über die gesamte Projektdauer. Die durch VCR generierten oder in VCR abgelegten Grafiken werden insbesondere für die *Project Status Decision (PSD) Meetings* verwendet, wo der Projektleiter monatlich den aktuellen Projektstatus und das weitere Vorgehen präsentieren muss.

Die nachfolgend gesammelten und visualisierten Beispiele bieten einen kleinen Überblick über häufig verwendete Grafiken, umfassen jedoch bei weitem nicht alle verfügbaren Visualisierungen.



Abbildung 11. Beispiele für aus VCR generierte Grafiken

### 1.3.3. Zukunft

Die Weiterentwicklung der aktuellen Version (2.x) von VCR soll bis auf weiteres erfolgen. Ob sich aus dem Resultat unserer Bachelorarbeit eine langfristig nachhaltige Lösung entwickelt, welche in wenigen Jahren VCR komplett ersetzen kann, kann zum jetzigen Zeitpunkt noch nicht gesagt werden.

Es bleibt festzuhalten, dass der Funktionsumfang von VCR beträchtlich ist und bereits heute das *Know How* aufgrund von personellen Wechslen nicht mehr überall komplett vorhanden ist. Zudem sind die personellen Ressourcen für eine Neuentwicklung aktuell stark limitiert. Aus diesen Gründen erachtet unser Auftraggeber eine *Big Bang* Ablösung von VCR, also eine nicht gestaffelte Komplettablösung, als wenig realistisch.

## 2. Projektmanagement

### Projektauftrag Bachelorarbeit 'Smart Enterprise Project Cockpit'

Studierende: Lukas Steiger, Pascal Hürlimann

Betreuer: Daniel Keller, HSR

Projektpartner: Thomas Weh, Siemens Schweiz AG, Zug

#### Einführung

Im Rahmen der Studienarbeit *"NextGen Project Management Tooling"* haben sich die Studierenden mit der bestehenden Projektmanagement Toollandschaft bei Siemens befasst. Dabei wurde zuerst eine Analyse der Arbeitsabläufe im Umfeld der Fortschritts- und Kostenverfolgung gemacht und eine Prototyp-Anwendung implementiert, welche einige *Pain Points* des aktuellen Ökosystems im Ansatz adressiert und entsprechende Handlungsfelder aufzeigt. Im Nachgang wurden die Resultate der Studienarbeit bei Siemens präsentiert und daraus abgeleitet der Projektauftrag für diese Bachelorarbeit definiert.

#### Aufgabenstellung

Bei Siemens gilt das langfristige Ziel, die Projektmanagementapplikation *"VCR"* Schritt für Schritt abzulösen. Im Rahmen dieser Bachelorarbeit soll nun ein erster Arbeitsablauf abgelöst und wenn möglich in den operativen Betrieb übergeben werden. Der Meilenstein Bewilligungsprozess ist ein wichtiger Prozess, in dessen Ablauf heute das Tool *VCR*, Excel, Email und händische Datenpflege gleichermaßen involviert sind. Im Rahmen der Arbeit soll auf Basis der Vorarbeit eine Nachfolgeapplikation für *VCR* aufgebaut werden, welche in einem ersten Schritt den Meilenstein Bewilligungsprozess digitalisiert. Bewährte Abläufe und Konzepte sollen in ähnlicher Form beibehalten werden. Das neue Tool soll als Grundlage dienen, um in Zukunft Stück für Stück weitere Funktionalität von *VCR* in die neue Lösung zu portieren.

## Erwartete Resultate

### Erwartetes Endprodukt

Das entwickelte Endprodukt digitalisiert den Meilensteinbewilligungsprozess und interagiert mit der bestehenden VCR Applikation. Der Datenaustausch und die Datenkonsistenz zwischen den beiden Tools werden in der durch die Studierenden zu entwickelnden Lösung umgesetzt.

### Abgabe

Termingerechte Abgabe des Berichts mit folgendem Inhalt (nicht abschliessend): Projektauftrag, Anforderungsanalyse, Software Architektur, Ergebnis, Fazit und Sitzungsprotokolle.

### Vorgehen

Die Studierenden...

- legen zu Beginn der Arbeit einen Terminplan mit Meilensteinen vor
- planen und entwickeln agil in enger Zusammenarbeit mit Betreuer und Projektpartner
- erfassen und pflegen ihre Arbeitspakete und legen zum Schluss eine Zeitauswertung vor

### Termine

#### 20 Februar 2019

Kick-off Sitzung

#### April/Mai 2019

Zwischenpräsentation - Termin wird noch festgelegt

**7 Juni 2019**

Freigabe Abstract

**14 Juni 2019**

Hochladen aller Dokumente und Abgabe des Berichts bis spätestens 12:00 Uhr.

**Juni-August 2019**

Mündliche BA Prüfung - Termin wird noch festgelegt

**Betreuung**

Wöchentliches Treffen (Mittwoch, 9:00) zur Standortbestimmung und zum gegenseitigen Austausch. Ansonsten Kontakt via Email oder auf Anfrage auch gern ein persönliches Treffen.

Rapperswil, 20.3.2019



Dr. Daniel Keller

## 2.1. Projektvision & Kontextabgrenzung

Im Rahmen einer Bachelorarbeit sollen Studierende mit realen Problemen konfrontiert werden, um sich damit für ihre spätere Berufslaufbahn zu qualifizieren. Dieser Grundsatz der HSR war auch uns bei der Wahl unserer Arbeit sehr wichtig. Sowohl für uns als auch für unseren Industriepartner Siemens war es zentral, dass wir am Ende ein Produkt abliefern können, welches einsatzfähig ist.

### 2.1.1. Projektvision

Zum Ende der Bachelorarbeit soll eine lauffähige und wenn möglich mit dem Produktivsystem verbundene Software ausgeliefert werden, welche erste ausgewählte Arbeitsabläufe aus dem alten Ökosystem im neu zu entwickelnden System abbildet.

#### Primäres Ziel

Der sogenannte Meilensteinbewilligungsprozess, welcher einer Abnahmeerklärung des aktuellen Projektstandes gleichkommt, soll in einer neuen Lösung umgesetzt werden, mit Anbindung an die bestehende VCR Software. Auf Basis bisheriger Erkenntnisse hat sich gezeigt, dass der Meilensteinbewilligungsprozess ein geeigneter Kandidat ist, um als erstes portiert zu werden. Denn:

- Der Projektleiter muss heute drei Tools (VCR, Excel und Outlook) nutzen. Mit der neuen Lösung muss er nur noch das neue Tool aktiv nutzen. Outlook soll neu nur noch als Erinnerungsmedium dienen.
- Die heute für die Meilensteinbewilligung im Einsatz stehende *Excel* Datei enthält sehr viel Logik und ist instabil. Zudem ist das *Know How* der Funktionsweise der *Excel* Datei aufgrund personeller Wechsel kaum mehr vorhanden.
- In den Meilensteinbewilligungsprozess sind zahlreiche Führungskräfte involviert. Eine gute Umsetzung ist eine Chance, um Sie von der Notwendigkeit einer Ablösung zu überzeugen.
- Der Arbeitsablauf ist gemäss aktuellem Kenntnisstand klar geregelt und die inhaltlichen Aspekte des bisherigen *Milestone Declaration Sheet* können übernommen werden.



Die nachfolgende Visualisierung der Projektmanagement Toollandschaft mit VCR als zentralem Dreh- und Angelpunkt wurde während der Studienarbeit entwickelt und nun für die Bachelorarbeit überarbeitet, adaptiert und der umzusetzende Teilaspekt wurde in *gelber Farbe* hervorgehoben.

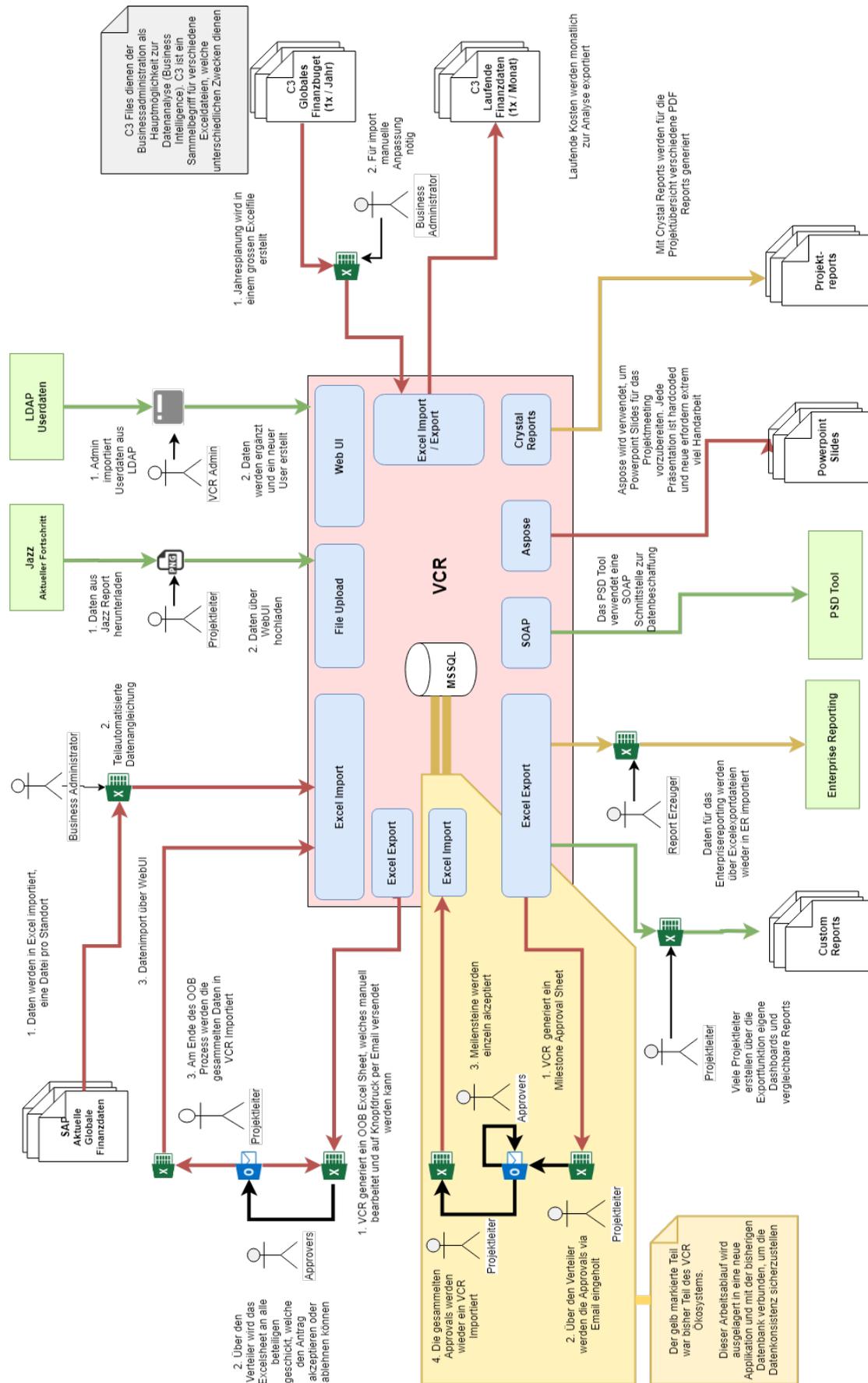


Abbildung 12. Mit dem primären Ziel, den Meilensteinbewilligungsprozess in einer neuen Lösung umzusetzen, lösen wir einen ersten Teil (Gelb markiert) aus dem bisherigen VCR Ökosystem ab.

## Sekundäre Ziele

Die Erreichung folgender Ziele würde zu einem grossen Mehrwert führen, deren Umsetzung hängt jedoch auch stark davon ab, wie aufwändig die Umsetzung des primären Ziels ist:

- Das Tool bietet personenbezogene, einfache Dashboards mit den wichtigsten Infos an (z.B. ausstehende Aufgaben).
- Ausarbeitung eines Konzepts für den Umgang mit historischen Daten. Wie können die Daten erfasst und die Anzeige von historischen Daten entlang der Zeitachse zurück in die Vergangenheit gesteuert werden.

### 2.1.2. Kontextabgrenzung

Während der Studienarbeit wurde die gesamte Toollandschaft im Bereich der Fortschritts- und Kostenverfolgung analysiert. Bei der Bachelorarbeit liegt der Fokus auf wenigen ausgewählten Funktionalitäten, welche heute nicht oder nur ungenügend in VCR integriert sind.

Andere Themengebiete, wie:

- Die Integration von anderen Tools oder Abläufen ins heutige VCR
- Schnittstellen zur Dateninteraktion für andere Tools
- Strategische Umsetzungspläne für die restliche Migration von VCR auf das neue Tool
- Analyse von Reporting Tools
- Umsetzung weiterer Geschäftsprozesse, z.B. Change Requests (OOB's)
- Implementieren von projektübergreifenden Trendsichten und Dashboards

sowie viele andere potenzielle Anforderungen, welche nicht durch die oben genannte Projektvision abgedeckt sind, können im Rahmen dieser Bachelorarbeit aus zeitlichen Gründen nicht angegangen werden.

## 2.2. Rahmenbedingungen, Artefakte & Lieferergebnisse

### 2.2.1. Zeitrahmen

Der vorgegebene Zeitrahmen pro Student beträgt 360 Stunden. Demzufolge stehen uns als Zweierteam für die Bachelorarbeit insgesamt 720 Stunden zur Verfügung.



Eine Zeitauswertung ist im Anhang unter [Metriken](#) zu finden.

### 2.2.2. Arbeitspaketverwaltung

Für die Verwaltung der Arbeitspakete hat sich während der Studienarbeit der Einsatz von IBM Rational Team Concert bewährt. Da die Instanz von Siemens zur Verfügung gestellt wird, können nach Beendigung der Bachelorarbeit sämtliche Arbeitspakete von einem zukünftigen Projektteam direkt weiterverwendet werden.

### 2.2.3. Quellcodeverwaltung

Die Resultate der vorangehenden Studienarbeit, der Quellcode der während der Bachelorarbeit entwickelten Software sowie die Dokumentation (*diese Bachelorarbeit, welche Sie gerade lesen*) sind auf GitHub abgelegt.

### 2.2.4. Applikationen & Tools

Die im Verlauf der Arbeiten in Betrieb genommenen Applikationen umfassen sowohl die Applikation selbst (ein Entwicklungssystem auf dem HSR Server sowie eine Produktivumgebung bei Siemens), automatisierte Build & Deployment Infrastruktur (*Jenkins*) sowie weitere Applikationen zur Verwaltung und Überwachung von Containern (*Portainer*) und zur Qualitätssicherung (*Sonar*).

### 2.2.5. Ergebnisse

Das aus der Bachelorarbeit resultierende Ergebnis ist später ausführlich im Kapitel [Ergebnisse & Ausblick](#) beschrieben.

## 2.3. Meilensteine

Zu Beginn der Bachelorarbeit haben wir zusammen mit Herrn Keller die nachfolgenden Meilensteine definiert. Da es der ausdrückliche Wunsch unseres Auftraggebers bei Siemens war, zum Ende der Bachelorarbeit ein *Roll-out* fähiges Endprodukt zu haben, welches vollumfänglich in die Siemens Infrastruktur integriert ist, haben wir uns für *Roll-out* getriebene Meilensteine entschieden. Wäre diese Vorgabe nicht gewesen, hätten wir Feature orientierte Pakete bevorzugt, wo wir bis zu einem bestimmten Zeitpunkt gewisse Teilfunktionalität hätten zur Verfügung stellen müssen.

### 2.3.1. Geplante Meilensteine

Die gleich nach dem Projektstart definierten Meilensteine waren die folgenden:

- |                         |               |   |
|-------------------------|---------------|---|
| <b>20. Februar 2019</b> | <i>[MS-0]</i> | Projektstart  |
| <b>20. März 2019</b>    | <i>[MS-1]</i> | Anforderungen an MS Approval Prozess geklärt, Architekturentwurf steht. Entwicklungsstart.                            |
| <b>17. April 2019</b>   | <i>[MS-2]</i> | Alphaversion des MS Approval Prozess bereit, Demo mit ausgewählten Stakeholdern durchgeführt und Feedback erhalten.   |
| <b>1. Mai 2019</b>      | <i>[MS-3]</i> | Betaversion des MS Approval Prozess. Deployment auf einen Dev-Server beim Kunden. Interaktion mit VCR Dev System.     |
| <b>15. Mai 2019</b>     | <i>[MS-4]</i> | Release Candidate. Testlauf mit ausgewählten Stakeholdern erneut durchgeführt und Feedback erhalten.                  |
| <b>29. Mai 2019</b>     | <i>[MS-5]</i> | Roll-out für ausgewählte Pilot-Projekte (Zugriffsbeschränkung). Software interagiert mit dem Produktivsystem von VCR. |
| <b>12. Juni 2019</b>    | <i>[MS-6]</i> | Abgabe Bachelorarbeit. Übergabe an Kunden.  |



Die obige Meilensteinliste ist unverändert seit 27. Februar. Es wurden lediglich nachträglich die "*[MS-X]*" Bezeichnungen hinzugefügt, um in der nachfolgenden Sektion darauf referenzieren zu können.

### 2.3.2. Erreichung der Meilensteine

Zum Ende der Bachelorarbeit haben wir die Zeilerreichung der einzelnen Meilensteine nochmals hinterfragt und nachfolgend einzeln ausgewiesen.

Tabelle 1. Zeilerreichung der einzelnen Meilensteine

Meilenstein	Status	Erläuterungen
[MS-0]	Erreicht	Der Projektstart ist pünktlich erfolgt.
[MS-1]	Erreicht	Die entsprechenden Punkte waren termingerecht zum 20. März erledigt. Auch wenn noch nicht alle Aspekte im Detail klar waren, wurde mit der Entwicklung gestartet.
[MS-2]	Erreicht	Exakt an diesem Tag war der erste "Durchstich" fertig, d.h. die allerersten, ausgewählten Daten wurden von VCR mit einem ETL Job in unsere Datenbank gelesen und konnten via Backend in der Benutzeroberfläche angezeigt werden. Eine erste informelle Demo wurde in der vorangehenden Woche mit dem <i>Operator</i> und dem <i>Product Owner</i> von VCR bei Siemens durchgeführt. Dabei haben wir Feedback erhalten und dieses später umgesetzt.
[MS-3]	Erreicht	Zu diesem Zeitpunkt konnten wir bereits mit den Daten des Produktivsystems von VCR arbeiten. Auch die Datenübertragung mit ETL hat bereits für einige Daten funktioniert.
[MS-4]	Teilweise erreicht	Am 14.5. hat unser Auftraggeber eine Demo der neuen Applikation im Rahmen eines Management Meetings mit interessierten Stakeholdern durchgeführt. Das Feedback war positiv, es fehlen allerdings noch einige Daten, welche zu diesem Zeitpunkt noch nicht gemappt sind. Das erhaltene Feedback wurde teilweise bereits umgesetzt, so war z.B. zum Zeitpunkt der Demo die Performanz der Projektliste ungenügend, dieser Aspekt wurde aber wenige Tage später behoben.
[MS-5]	Teilweise erreicht	Die Software integriert mit dem Produktivsystem von VCR, es fehlt aber noch an einigen Dingen, um einen Roll-out für gewisse Pilotprojekte zu machen. Genaueres ist dem Kapitel <a href="#">Zielerreichung &amp; Umsetzungsgrad</a> zu entnehmen.
[MS-6]	Erreicht	Der Abgabetermin der Bachelorarbeit ist zwingend und wird deshalb auch eingehalten. Die Abgabe erfolgt am 11. Juni, die Übergabe der Resultate an Siemens erfolgt einige Tage später.

## 3. Anforderungsanalyse

### 3.1. Einführung

Während der Studienarbeit wurden die Schwächen und Stärken der bestehenden Landschaft analysiert und Massnahmen zur Umsetzung definiert. Als Hauptmassnahme wurde eine mittelfristige Ablösung von VCR durch eine neue Lösung definiert. Der Zielstand des Funktionsumfangs der neuen Applikation beinhaltet unter anderem folgende Funktionalität:

- *Projektverwaltung mit zentralen Metadaten*
- *Abbildung des Meilenstein Bewilligungsprozess*
- *Change Requests für Budget und Zeitplan*
- *Persönliche Dashboards*
- *Gefilterte und aggregierte Projektsichten*
- *Zugriff auf Daten über API*
- *Aufbereitung der Daten für die noch zu evaluierende Reporting Lösung*
- *Exports zu Excel*

— [\[SA-21\]](#)

Nebst den soeben zitierten funktionsorientierten Anforderungen ist den folgenden nichtfunktionalen Aspekten grosses Gewicht zu schenken:

- *Die Applikation soll auf diversen Endgeräten (Mobiltelefone, Tablets, Laptops und Desktop PC's) genutzt werden können. Der zur Verfügung stehende Raum soll dabei jeweils optimal ausgenutzt werden.*
- *Eine flüssige Bedienung ist sehr wichtig, da die Applikation auch von vielen Mitarbeitenden mit Führungsfunktion genutzt wird. Deren Zeit ist knapp und kostbar für die Firma.*
- *Bei der Wahl von Dritthersteller Softwarekomponenten (Programmiersprachen, Runtimes, Libraries, Datenbanksysteme) sollen Aspekte wie Lizenz- und Betriebskosten, deren Zukunftsträchtigkeit und deren Lizenzierungsmodell genau angeschaut und abgestimmt werden.*

— [\[SA-21\]](#)

Weiter haben wir einige zentrale [Qualitätsziele](#) definiert und uns mit unseren [\[Stakeholder\]n](#) auseinandergesetzt.

Wir haben eine [Nutzungsanalyse](#) durchgeführt, wo wir mithilfe von ausführlichen [Interviews](#) die zentralen [Probleme](#) identifiziert haben. Nachfolgend sind noch einmal die wichtigsten Erkenntnisse zusammengefasst:

- Viele Probleme sind durch das historische Wachstum von VCR bedingt.
- Es gibt eine [Schlechte Toolintegration](#) mit Drittsystemen.
- Das bestehende [Rechtemanagement](#) in der Applikation erfüllt seinen Zweck nur ungenügend.
- Es gibt eine starke [Notwendigkeit für Excel](#), da viele Workflows ungenügend ins Tool integriert sind oder teilweise auch auf Excel basieren.
- Ein [Mühsamer Datenimport](#) sorgt dafür, dass das Einspeisen von externen Daten zur Geduldsprobe wird.
- Die [Schlechte Performance](#) der Applikation äussert sich z.B. in sehr langen Ladezeiten bei diversen Vorgängen.

In einem nächsten Schritt haben wir in einer [Bestandsanalyse](#) die wichtigsten Funktionen von [VCR](#) analysiert sowie zwei der wichtigsten Workflows, welche heute hauptsächlich ausserhalb von VCR abgewickelt werden, aufgezeigt. Den einen Workflow, den [Meilensteinbewilligungsprozess](#), haben wir uns im Rahmen unserer Bachelorarbeit vorgenommen. Dieser ist im nachfolgenden Kapitel beschrieben. Der andere Prozess, um Budget oder Terminänderungen beglaubigen zu lassen ([Budget & Schedule Change Request \(OOB\) Sheet](#)), bietet sich an, um als nächstes in die neue Lösung portiert zu werden.

Mithilfe von [Personas](#) und [Benutzungsszenarien](#) haben wir zudem noch eine andere Sichtweise auf die Anforderungen an die neue Applikation erstellt.

### 3.2. Meilensteinbewilligungsprozess

So gut wie jeder Projektmanagementprozess kennt Meilensteine. Sie dienen als Etappenziele innerhalb eines oft langläufigen Projektverlaufs. Typischerweise gibt der entsprechende Geschäftsprozess dabei vor, welche Lieferergebnisse zu welchem Meilenstein erwartet werden. Die bisher investierten Projektkosten, das geplante Restbudget sowie die Termineinhaltung, z.B. mit einer Meilensteintrendanalyse, sind dabei ein wichtiger Bestandteil. Gerade in Grossfirmen wie Siemens ist es typisch, dass verschiedene Personen, welche jeweils eine andere Rolle im Kontext des jeweiligen Projektes innehaben, zur Zielerreichung des jeweiligen Meilensteins ihre explizite Zustimmung geben müssen. Diese Entscheidung muss aus Gründen der Beweisbarkeit und Rückverfolgbarkeit schriftlich erfolgen.

#### 3.2.1. Bisheriger Meilensteinbewilligungsprozess

Der Meilensteinbewilligungsprozess, so wie er heute bei [\[Siemens\]](#) praktiziert wird, lässt sich am einfachsten anhand folgender Grafik erläutern:

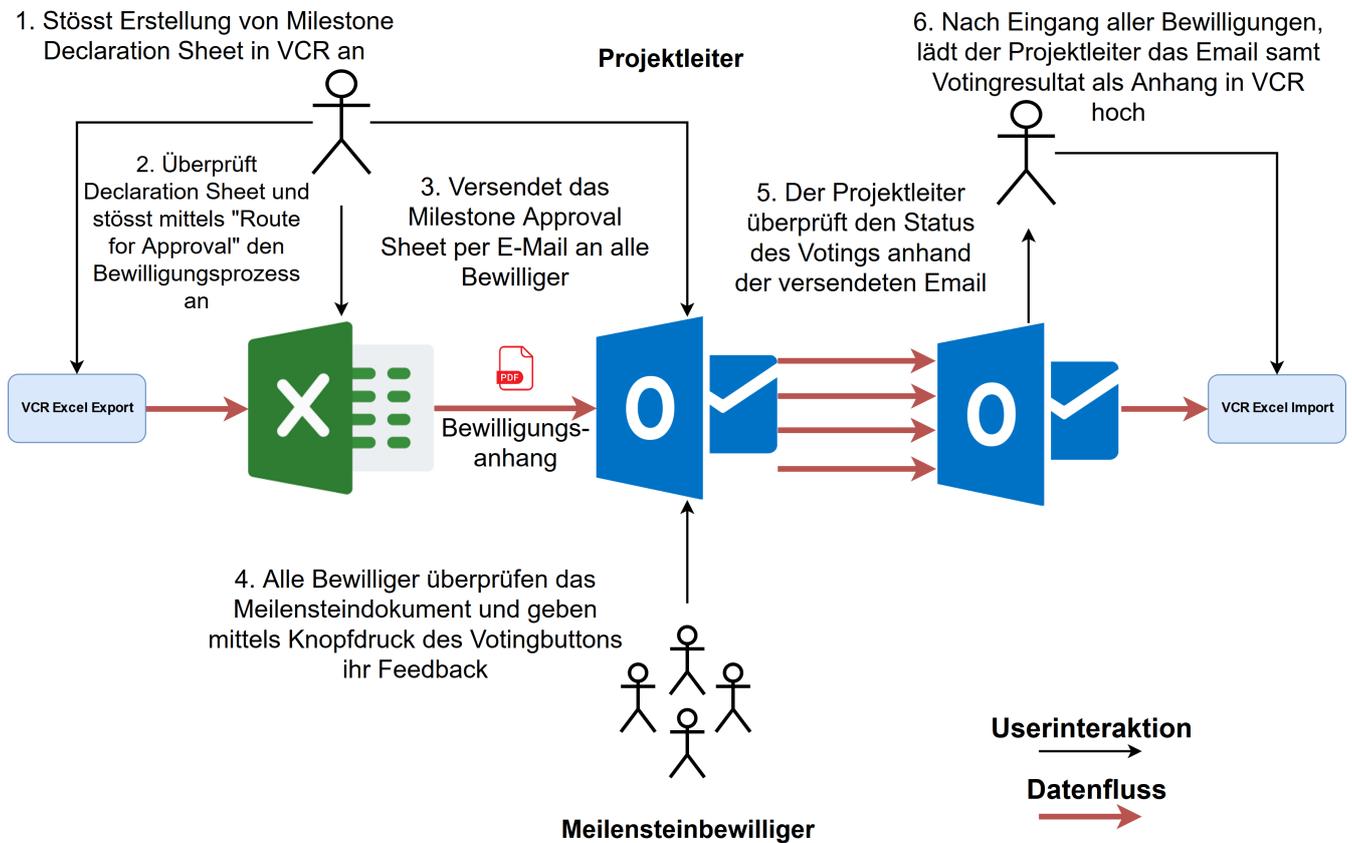


Abbildung 13. Bisheriger Meilensteinbewilligungsprozess bei [\[Siemens\]](#)

Die einzelnen Schritte des oben dargestellten Bewilligungsprozesses sind auf den nachfolgenden Seiten im Detail erläutert.

*Exkurs: Meilensteintrendanalyse*

Voraussetzung für eine *Meilensteintrendanalyse* ist, dass für ein Projekt mehrere Meilensteine definiert sind und diese mit einem Zieldatum zur Erreichung des Meilensteins versehen sind. Im Verlauf des Projektes müssen zudem jeweils regelmässig die ursprünglich versprochenen und aktuell noch nicht erreichten Meilensteintermine hinterfragt und allenfalls angepasst werden. Die Meilensteintrendanalyse macht sich die zeitliche Verschiebung von Meilensteinen zu Nutze, um eine Prognose für die weiteren Projekttermine zu erhalten. Die Grafik ist dabei ein besonders wirkungsvolles Instrument, denn aus den Linien lässt sich auf einen Blick ein Trend ablesen.

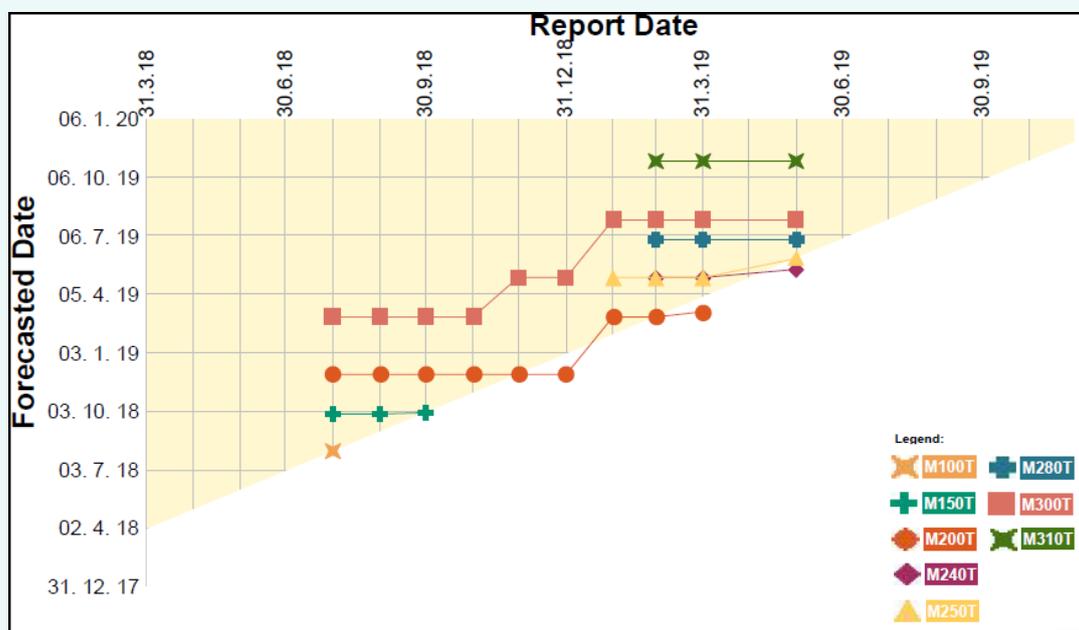


Abbildung 14. Beispiel einer Meilensteintrendanalyse aus einem realen Projekt bei Siemens zum Betrachtungszeitpunkt am 31. Mai 2019.

- Horizontale Linien bedeuten, dass die Termine wie geplant eingehalten werden.
- Steigende Linien bedeuten eine Verzögerung des Meilensteintermins.
- Fallende Linien bedeuten eine frühere Fertigstellung.

Wenn frühzeitig Trends erkannt werden, können ggf. Gegenmaßnahmen zur Einhaltung von Terminen eingeleitet werden.

— [\[MTA-einfach-erklärt\]](#) - empfehlenswerter Artikel zu diesem Thema

## Schritt 1: Erzeugung des Declaration Sheets

Als Projektverantwortlicher ist es die Aufgabe des Projektleiters, den Prozess der Meilensteinbewilligung zu starten. Bisher ging er dazu ins *VCR* und navigierte dort zum Zeitplan (*Schedule*). Dort war es bisher möglich, für sämtliche Meilensteine ein *Milestone Declaration Sheet* zu generieren. Neu soll dieser Prozess nur noch für den aktuellen Meilenstein möglich sein, da dieser Prozess nur für den aktuell anstehenden Meilenstein Sinn ergibt.

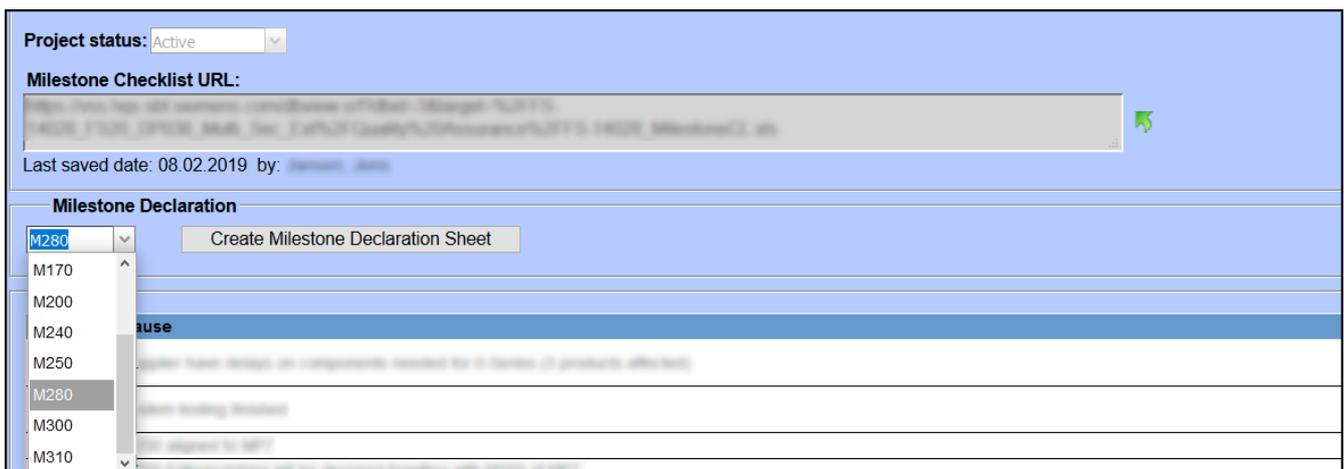


Abbildung 15. Heutige Ansicht zur Erzeugung eines Milestone Declaration Sheets (anonymisiert)

## Schritt 2: Überprüfung der Daten

Als nächstes wird die heruntergeladene Datei geöffnet. Der Projektleiter prüft die Kennzahlen und kümmert sich um die Warnungen (Status = *red*). Solange diese Abweichungen nicht korrigiert sind, kann der Bewilligungsprozess nicht gestartet werden. Bei zu grossen Budget- oder Terminabweichungen muss zuerst ein zugehöriger Change Request (OOB) gemacht werden, welcher die Budget- und/oder Terminüberschreitung legitimiert.

Das gesamte Milestone Declaration Sheet ist auf den folgenden Seiten zu finden.

M240 Declaration for										Indicator																																																																																																				
<b>Milestones to declare for process-variant PEP-Agile</b> M100 M120 M150 M170 M200A M240 M250 M280 M300 M310 x										red																																																																																																				
<b>Project Data</b> Project Name: ... Group ID / Loc. / Project ID: ... Product Line: ... Process Variant: ... Quality Gate: ... Milestone Date: 12.10.2018 PL: ... PM: ... Project Quality Manager: ... PPL: ... PPM: ...										Status Milestone Readiness																																																																																																				
<b>Milestone Readiness</b> MS Readiness Check done?																																																																																																														
<b>Requirements and Features</b> <table border="1"> <thead> <tr> <th></th> <th></th> <th>Must Req. Prio 0   Prio 1</th> <th>Value Req. Prio 2   Prio 3</th> <th>Bugfix</th> <th>Total</th> <th>USP</th> </tr> </thead> <tbody> <tr> <td>M150</td> <td>Initial Requirements (PM)</td> <td>221   87</td> <td>13   0</td> <td>33</td> <td>354</td> <td>0</td> </tr> <tr> <td>M200A</td> <td>Resulting Requirements (PM+PL)</td> <td>194   39</td> <td>4   0</td> <td>23</td> <td>260</td> <td>0</td> </tr> <tr> <td>M240</td> <td>Resulting Requirements (PL)</td> <td>0   0</td> <td>0   0</td> <td>0</td> <td></td> <td>0</td> </tr> <tr> <td colspan="7">Sum(Prio 0, Prio 1, Prio 2) == 90% of Sum(Prio 0, Prio 1, Prio 2)@M28</td> </tr> </tbody> </table>												Must Req. Prio 0   Prio 1	Value Req. Prio 2   Prio 3	Bugfix	Total	USP	M150	Initial Requirements (PM)	221   87	13   0	33	354	0	M200A	Resulting Requirements (PM+PL)	194   39	4   0	23	260	0	M240	Resulting Requirements (PL)	0   0	0   0	0		0	Sum(Prio 0, Prio 1, Prio 2) == 90% of Sum(Prio 0, Prio 1, Prio 2)@M28							red Requirements																																																																	
		Must Req. Prio 0   Prio 1	Value Req. Prio 2   Prio 3	Bugfix	Total	USP																																																																																																								
M150	Initial Requirements (PM)	221   87	13   0	33	354	0																																																																																																								
M200A	Resulting Requirements (PM+PL)	194   39	4   0	23	260	0																																																																																																								
M240	Resulting Requirements (PL)	0   0	0   0	0		0																																																																																																								
Sum(Prio 0, Prio 1, Prio 2) == 90% of Sum(Prio 0, Prio 1, Prio 2)@M28																																																																																																														
<b>Budget / Cost (MEURO)</b> 1. Overall Total Project Budget: ... Total Forecast: ... Forecast Threshold for OOB-Notification: ... OOB required -- Forecast exceeds AdjustedProjectBudget by ... 2. Major Project Phases All actuals refer to SAP bookings + Obligos until: 2018-09-30 Cost Limit until M200A: ... Cost booked until M200A: ... Cost budgeted for Phase M200A to M310: ... Cost Forecast for Phase M200A to M310: <table border="1"> <thead> <tr> <th>@M100</th> <th>M100-M120</th> <th>M120-M150</th> <th>M150-M170</th> <th>M170-M200A</th> <th>M200A-M240</th> <th>M240-M250</th> <th>M250-M300</th> <th>after M300</th> </tr> </thead> <tbody> <tr> <td>30</td> <td>insert   131</td> <td>insert   1635</td> <td>insert   1036</td> <td>insert   1404</td> <td>insert   9039</td> <td>insert   6876</td> <td>insert   1845</td> <td>insert   0</td> </tr> <tr> <td>Actual</td> <td>Baseline</td> <td>Actual</td> <td>Baseline</td> <td>Actual</td> <td>Baseline</td> <td>Actual</td> <td>Baseline</td> <td>Forecast</td> </tr> <tr> <td>@M100</td> <td>@M100</td> <td>@M100</td> <td>@M150</td> <td>@M150</td> <td>@M200A</td> <td>@M200A</td> <td>@M200A</td> <td>@M200A</td> </tr> </tbody> </table>										@M100	M100-M120	M120-M150	M150-M170	M170-M200A	M200A-M240	M240-M250	M250-M300	after M300	30	insert   131	insert   1635	insert   1036	insert   1404	insert   9039	insert   6876	insert   1845	insert   0	Actual	Baseline	Actual	Baseline	Actual	Baseline	Actual	Baseline	Forecast	@M100	@M100	@M100	@M150	@M150	@M200A	@M200A	@M200A	@M200A	Alerts Yes OOB Alert on Total Budget No EPP Alert on Total Budget																																																																
@M100	M100-M120	M120-M150	M150-M170	M170-M200A	M200A-M240	M240-M250	M250-M300	after M300																																																																																																						
30	insert   131	insert   1635	insert   1036	insert   1404	insert   9039	insert   6876	insert   1845	insert   0																																																																																																						
Actual	Baseline	Actual	Baseline	Actual	Baseline	Actual	Baseline	Forecast																																																																																																						
@M100	@M100	@M100	@M150	@M150	@M200A	@M200A	@M200A	@M200A																																																																																																						
3. Business Year previous BYs: ... BY-Budget: ... BY-Forecast: ... BY-Actuals + Obligos: ... For future BYs: ... OOB required -- BY19-Forecast exceeds BY budget by ...										Yes-(Glob) OOB Alert on BY Budget																																																																																																				
<b>Schedule (MYA: see Project Report)</b> <table border="1"> <thead> <tr> <th>M100</th> <th>M120</th> <th>M150</th> <th>M170</th> <th>M200A</th> <th>M240</th> <th>M250</th> <th>M280</th> <th>M300</th> <th>M310</th> </tr> </thead> <tbody> <tr> <td>Jul-15</td> <td>Oct-15</td> <td>Oct-16</td> <td>Mar-17</td> <td>Jul-17</td> <td>Oct-18</td> <td>Jun-19</td> <td>Oct-19</td> <td>Nov-19</td> <td>Dec-19</td> </tr> <tr> <td>reached</td> <td>reached</td> <td>reached</td> <td>reached</td> <td>reached</td> <td>scheduled</td> <td>planned</td> <td>planned</td> <td>planned</td> <td>planned</td> </tr> </tbody> </table> <b>Milestone Trend History</b> <table border="1"> <thead> <tr> <th>@M100 &gt;&gt;&gt;</th> <th>@M120 &gt;&gt;&gt;</th> <th>@M150 &gt;&gt;&gt;</th> <th>@M170 &gt;&gt;&gt;</th> <th>@M200A &gt;&gt;&gt;</th> <th>@M240 &gt;&gt;&gt;</th> <th>@M250 &gt;&gt;&gt;</th> <th>@M280 &gt;&gt;&gt;</th> <th>@M300 &gt;&gt;&gt;</th> <th>@M310 &gt;&gt;&gt;</th> </tr> </thead> <tbody> <tr> <td>Sep-15</td> <td>Dec-15</td> <td>Jun-16</td> <td>Jun-17</td> <td>Oct-18</td> <td>Jan-19</td> <td>Jul-19</td> <td>Dec-17</td> <td>Dec-17</td> <td>Dec-19</td> </tr> <tr> <td></td> <td>Feb-16</td> <td>Aug-16</td> <td>Jun-17</td> <td>Oct-18</td> <td>Jun-19</td> <td>Jul-19</td> <td>Dec-17</td> <td>Dec-17</td> <td>Dec-19</td> </tr> <tr> <td></td> <td></td> <td>Jun-17</td> <td>Jun-17</td> <td>Oct-18</td> <td>Jun-19</td> <td>Jul-19</td> <td>Sep-17</td> <td>Sep-17</td> <td>Sep-19</td> </tr> <tr> <td></td> <td></td> <td>Feb-17</td> <td>Jun-17</td> <td>Oct-18</td> <td>Jun-19</td> <td>Jul-19</td> <td>Sep-19</td> <td>Sep-19</td> <td>Sep-19</td> </tr> <tr> <td></td> <td></td> <td></td> <td>Jun-17</td> <td>Oct-18</td> <td>Jun-19</td> <td>Jul-19</td> <td>Jul-19</td> <td>Jul-19</td> <td>Oct-19</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>Oct-18</td> <td>Jun-19</td> <td>Jul-19</td> <td>Oct-19</td> <td>Nov-19</td> <td>Dec-19</td> </tr> </tbody> </table>										M100	M120	M150	M170	M200A	M240	M250	M280	M300	M310	Jul-15	Oct-15	Oct-16	Mar-17	Jul-17	Oct-18	Jun-19	Oct-19	Nov-19	Dec-19	reached	reached	reached	reached	reached	scheduled	planned	planned	planned	planned	@M100 >>>	@M120 >>>	@M150 >>>	@M170 >>>	@M200A >>>	@M240 >>>	@M250 >>>	@M280 >>>	@M300 >>>	@M310 >>>	Sep-15	Dec-15	Jun-16	Jun-17	Oct-18	Jan-19	Jul-19	Dec-17	Dec-17	Dec-19		Feb-16	Aug-16	Jun-17	Oct-18	Jun-19	Jul-19	Dec-17	Dec-17	Dec-19			Jun-17	Jun-17	Oct-18	Jun-19	Jul-19	Sep-17	Sep-17	Sep-19			Feb-17	Jun-17	Oct-18	Jun-19	Jul-19	Sep-19	Sep-19	Sep-19				Jun-17	Oct-18	Jun-19	Jul-19	Jul-19	Jul-19	Oct-19					Oct-18	Jun-19	Jul-19	Oct-19	Nov-19	Dec-19	Alerts Yes OOB Alert on Schedule Yes EPP Alert on Schedule
M100	M120	M150	M170	M200A	M240	M250	M280	M300	M310																																																																																																					
Jul-15	Oct-15	Oct-16	Mar-17	Jul-17	Oct-18	Jun-19	Oct-19	Nov-19	Dec-19																																																																																																					
reached	reached	reached	reached	reached	scheduled	planned	planned	planned	planned																																																																																																					
@M100 >>>	@M120 >>>	@M150 >>>	@M170 >>>	@M200A >>>	@M240 >>>	@M250 >>>	@M280 >>>	@M300 >>>	@M310 >>>																																																																																																					
Sep-15	Dec-15	Jun-16	Jun-17	Oct-18	Jan-19	Jul-19	Dec-17	Dec-17	Dec-19																																																																																																					
	Feb-16	Aug-16	Jun-17	Oct-18	Jun-19	Jul-19	Dec-17	Dec-17	Dec-19																																																																																																					
		Jun-17	Jun-17	Oct-18	Jun-19	Jul-19	Sep-17	Sep-17	Sep-19																																																																																																					
		Feb-17	Jun-17	Oct-18	Jun-19	Jul-19	Sep-19	Sep-19	Sep-19																																																																																																					
			Jun-17	Oct-18	Jun-19	Jul-19	Jul-19	Jul-19	Oct-19																																																																																																					
				Oct-18	Jun-19	Jul-19	Oct-19	Nov-19	Dec-19																																																																																																					
<b>Economic Data</b> There is an EPP Alert or an undefined EPP state. Please update EPP and in VCR the EPP Data and/or planning data first. 1. General EPP related data: Profitability of this project has been calculated in an EPP? Yes EPP belongs to a different project? No IMPORTANT: EPP update required! IMPORTANT: OOB on Schedule required. New data may require EPP update. IMPORTANT: OOB on Budget required. New data may require EPP update.										Alerts Yes Collective EPP Alert																																																																																																				

Abbildung 16. Beispiel eines Milestone Declaration Sheets (anonymisiert - oberer Teil)

2. Product data:						
<b>System Controller PXC5.00</b>						
Current	SHK	EUR	Volumes Year1	Volumes Year2	Volumes Year3	No OOB Alert
Target	@M150	EUR				No EPP Alert
Projected	@M200A	EUR				
Last Approved EPP (Last Confirmed)						
<b>Equipment Controller PXC4.E</b>						
Current	SHK	EUR	Volumes Year1	Volumes Year2	Volumes Year3	No OOB Alert
Target	@M150	EUR				No EPP Alert
Projected	@M200A	EUR				
Last Approved EPP (Last Confirmed)						
<b>TX-IO 4 UIO / 3 DO</b>						
Current	SHK	EUR	Volumes Year1	Volumes Year2	Volumes Year3	No OOB Alert
Target	@M150	EUR				No EPP Alert
Projected	@M200A	EUR				
Last Approved EPP (Last Confirmed)						
<b>Remarks / Process Exceptions</b>						
<b>Approvals</b>						
<b>1. Management Approval:</b> Please note: The local LOA or signature regulation may override the given list of approvers below.						<b>Electronic Signature</b>
Segment Head *)	Name					<input checked="" type="checkbox"/>
Segment BA **)	Name					<input checked="" type="checkbox"/>
Global RD **)	Name					<input checked="" type="checkbox"/>
Next Level RD BA *)	Name					<input checked="" type="checkbox"/>
RD BA	Name					<input checked="" type="checkbox"/>
Next Level PPM *)	Name					<input checked="" type="checkbox"/>
Next Level PPL *)	Name					<input checked="" type="checkbox"/>
PPL	Name					<input checked="" type="checkbox"/>
PPM	Name					<input checked="" type="checkbox"/>
PL	Name					<input checked="" type="checkbox"/>
PM	Name					<input checked="" type="checkbox"/>
Head System Test	Name					<input checked="" type="checkbox"/>
Project Quality Manager	Name					<input checked="" type="checkbox"/>
<b>2. Additional Approval ***) :</b>						
Function / Role	Name	Date	Signature			<input type="checkbox"/>
*) Management approval mandatory for projects with Total Project Budget or Total Forecast!						
**) Segment BA approval mandatory for projects with Total Project Budget or Total Forecast!						

Abbildung 17. Beispiel eines Milestone Declaration Sheets (anonymisiert - unterer Teil)



### Schritt 3: Bewilligungsprozess auslösen

Sobald alle Angaben im Sinne des Projektleiters sind und die beiden Eingabefelder 'Milestone Readiness Check done?' und 'Remarks / Process Exceptions' ausgefüllt sind, kann der Projektleiter entweder:

- Über die Vorschau den Inhalt des generierten PDF prüfen.
- Den Approval Prozess auslösen, indem er über die *Route for Approval* Funktion eine Email mit Verteilerliste, dem PDF als Anhang und den integrierten Voting Buttons generiert.

### Schritt 4-6: Ablauf des Bewilligungsprozess

Nach der Erzeugung der Email aus Excel läuft der Prozess wie folgt ab:

- Die automatisch erzeugte Email mit Voting Buttons (Approve, Approve with comments, decline) kann jetzt alle notwendigen Approver verschickt werden.
- Der Sender MUSS dabei die Email im *Sent* Order belassen, sobald diese verschickt wurde. Ansonsten können die *Voting* Resultate nicht gesammelt werden.
- Die versendete Email enthält die konsolidierte Liste des *Votings*. Somit ist jederzeit in der gesendeten Email ersichtlich, wie der Status des *Votings* ist.
- Wenn sich alle Approver die erhaltene Email (mit dem Approval PDF als Anhang) angesehen haben und Ihr Feedback (Approved, Approved with Comments, Decline) via Voting Buttons ausgedrückt haben, ist der Gesamtstatus in der gesendeten Mail sichtbar.
- Sobald sämtliche Rückmeldungen beim Versender eingegangen sind, lädt der Sender das Email mit den Voting Informationen in VCR hoch und markiert den Meilenstein in VCR als erreicht.

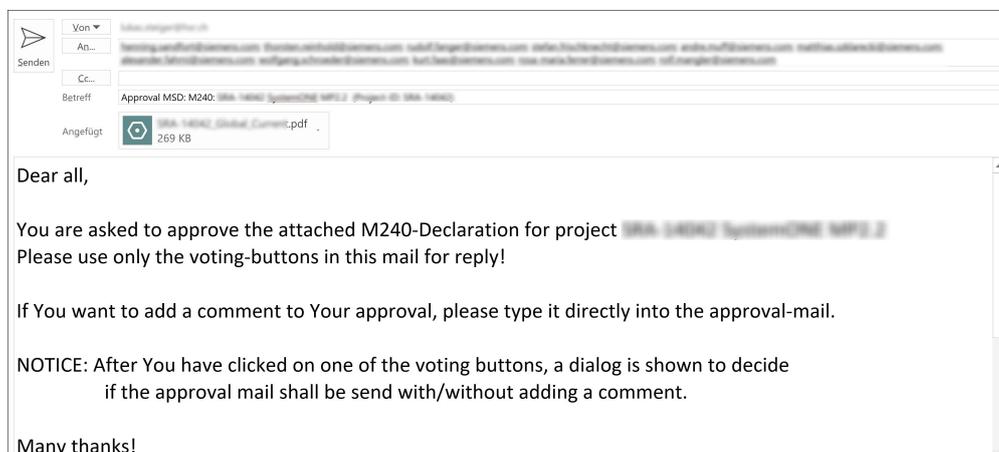


Abbildung 20. Beispiel einer generierten Approval Email

### 3.2.2. Analyse der bisherigen Umsetzung

Um eine neue Lösung umsetzen zu können, welche eine bisherige Lösung ersetzen und verbessern soll, musste zuerst die heutige Lösung detailliert analysiert werden. In Bezug auf den Workflow wurde dies bereits im vorangehenden Kapitel gemacht. Wie allerdings die dargestellten Daten zu Stande kommen und wie die in der Excel Datei eingebaute Visualisierungs- und Geschäftslogik funktioniert, war bislang unbekannt. Um sicherzustellen, dass uns die dort genutzten Daten überhaupt zur Verfügung stehen, wurde in mehreren Schritten das gesamte bisherige *Milestone Declaration Sheet* bis ins kleinste Detail analysiert.

#### Datenquelle & Datenspeicherung

Das *Milestone Declaration Sheet* mit all seinen Funktionen ist als Template in VCR abgelegt. Sobald der Erstellungsprozess der Excel Datei gestartet wird, wird serverseitig das Excel Template geöffnet und die berechneten Daten werden auf einer versteckten Arbeitsmappe abgelegt. Das bedeutet, dass sämtliche Daten bereits *offline* in der Excel Datei vorhanden sind.



Bei genauerer Durchsicht dieser Daten und einem Vergleich mit denen in der VCR Datenbank gespeicherten Daten wurde klar, dass man es mit sehr viel vorberechneten Daten zu tun hat.

Die Berechnung dieser Daten ist unausweichlich, da es sich oft um aggregierte Werte handelt. Zur Berechnung dieser Daten standen zwei Lösungsansätze zur Diskussion:

- Es werden nur Daten aus der Datenbank genutzt und die vorberechneten, aggregierten Daten werden in der neuen Lösung berechnet.
- Es wird weiterhin auf VCR gesetzt, um an die vorberechneten Daten zu kommen.

Die erste Variante hat zwei gewichtige Nachteile: Sie ist sehr zeitaufwändig, da sämtliche Berechnungen neu implementiert werden müssen. Zudem führt sie zu duplizierter Geschäftslogik, da die berechneten Daten auch weiterhin in VCR benötigt werden.

Somit ist es unumgänglich, dass uns die VCR Entwickler die entsprechenden Daten über eine dedizierte Schnittstelle zur Verfügung stellen müssen.

#### Visualisierungslogik

Hinter praktisch jedem Feld im alten *Milestone Declaration Sheet* steckt eine Formel, welche für die korrekte Anzeige des entsprechenden Wertes sorgt.

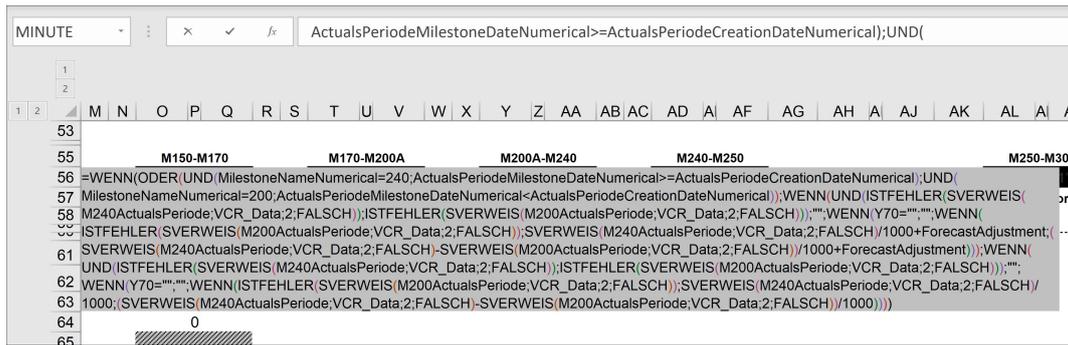


Abbildung 21. Beispiel einer Feldformel aus dem Milestone Declaration Sheet.

In den meisten Fällen ist dies vor allem Anzeigelogik, welche dafür sorgt, dass je nach Meilenstein und je nachdem, welche Daten zur Verfügung stehen, andere Information angezeigt wird. Diese Anzeigelogik müssen wir in unserer neuen Applikation selbst umsetzen. Um dies tun zu können und zu verstehen, ob die entsprechenden Daten über die neue VCR Schnittstelle für berechnete Daten zur Verfügung stehen, haben wir eine umfangreiche [\[Schnittstellenanalyse\]](#) durchgeführt, welche im Anhang unter [\[Schnittstellenanalyse\]](#) zu finden ist.

### Geschäftslogik

An einigen Orten ist auch Geschäftslogik vorhanden. Diese muss ebenfalls in der neuen Lösung nachgebaut werden, um den Meilensteinbewilligungsprozess ablösen zu können. Bei der Analyse für Überraschung gesorgt hat, dass die sogenannte Approval Matrix nicht Teil von VCR ist. Die Approval Matrix spezifiziert, welche Projektrollen zu welchem Meilenstein eine Unterschrift geben müssen. Eine solche Approval Matrix gibt es dabei jeweils einmal pro Prozesstyp. Diese Approval Matrix ist lediglich in der generierten Excel Datei zu finden und muss in Folge dessen ebenfalls in der neuen Lösung umgesetzt werden.

Conditional approvers:		Implemented logic as follows:									
*) approval not mandatory if TotalProjectBudget < 500kEuro		Table shows the default settings for projects which don't fulfill any given app									
**) approval not mandatory if TotalProjectBudget < 500kEuro		In case that a condition is fulfilled, for conditional approvers a TRUE (=WAHR) will be replaced by a FALSE (=FALSCH); FALSE itself stays									
Default settings for MSD approval by role for PEP and PEP Agile Projects:											
	M100	M120	M150	M170	M200	M240	M250	M280	M300	M310	
Segment Head *)	WAHR	WAHR	WAHR	FALSCH	WAHR	WAHR	WAHR	WAHR	WAHR	FALSCH	
Segment BA *)	FALSCH	FALSCH	WAHR	FALSCH	WAHR	WAHR	WAHR	WAHR	WAHR	FALSCH	
Next Level RD BA **)	FALSCH	FALSCH	WAHR	FALSCH	WAHR	WAHR	WAHR	WAHR	FALSCH	WAHR	
Next Level PPM **)	WAHR	WAHR	WAHR	FALSCH	WAHR	WAHR	WAHR	WAHR	WAHR	FALSCH	
Next Level PPL **)	WAHR	WAHR	WAHR	FALSCH	WAHR	WAHR	WAHR	WAHR	WAHR	FALSCH	
Head Global RD **)	FALSCH	FALSCH	WAHR	FALSCH	WAHR	WAHR	WAHR	FALSCH	WAHR	FALSCH	
RD BA	WAHR	WAHR	WAHR	FALSCH	WAHR	WAHR	WAHR	FALSCH	WAHR	WAHR	
PPM	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	
PPL	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	
PM	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	
PL	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	
PQM	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	
Line Manager System Test	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	WAHR	WAHR	WAHR	WAHR	FALSCH	
Head of Production	FALSCH	FALSCH	FALSCH	FALSCH	WAHR	FALSCH	WAHR	WAHR	WAHR	FALSCH	
Segment QM	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	
PLM Controller	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	

Abbildung 22. Approval Matrix für den Prozesstyp PEP.

### 3.2.3. Neuer Bewilligungsprozess

Der vorangehend erläuterte Meilensteinbewilligungsprozess sowie die detaillierte Analyse der bisherigen Umsetzung haben einige Schwachstellen aufgezeigt, welche mit einer neuen Lösung eliminiert werden sollen. Zudem waren einige Schwachstellen bereits bei seit der [Nutzungsanalyse](#) und der [Bestandsanalyse](#) bekannt.

Nach reiflicher Überlegung und diversen Gesprächen mit unseren Stakeholdern haben wir uns für eine einfache und pragmatische Lösung entschieden. Der gesamte Ablauf soll innerhalb des neuen Tools geschehen. Das einzige Werkzeug, welchem ausserhalb unserer Lösung noch eine Bedeutung zukommt, ist das Email Programm der Anwender. Es ist eine Realität, dass der überwältigende Anteil der Mitarbeiter heute noch sehr Email getrieben arbeiten. Dies ist auch absolut legitim, da es diverse Programme gibt, mit welchen Sie regelmässig arbeiten und wo Sie Aufgaben abzarbeiten haben. Der Posteingang des Email Kontos ist dabei üblicherweise der Dreh- und Angelpunkt für Erinnerungen, Termine, ausstehende Aufgaben und Konversationen. Aus diesem Grund möchten wir weiterhin Emails als Erinnerungs- und Kommunikationsmedium nutzen, um unsere Anwender auf ausstehende Aufgaben (in diesem Fall *Approvals*) hinzuweisen.

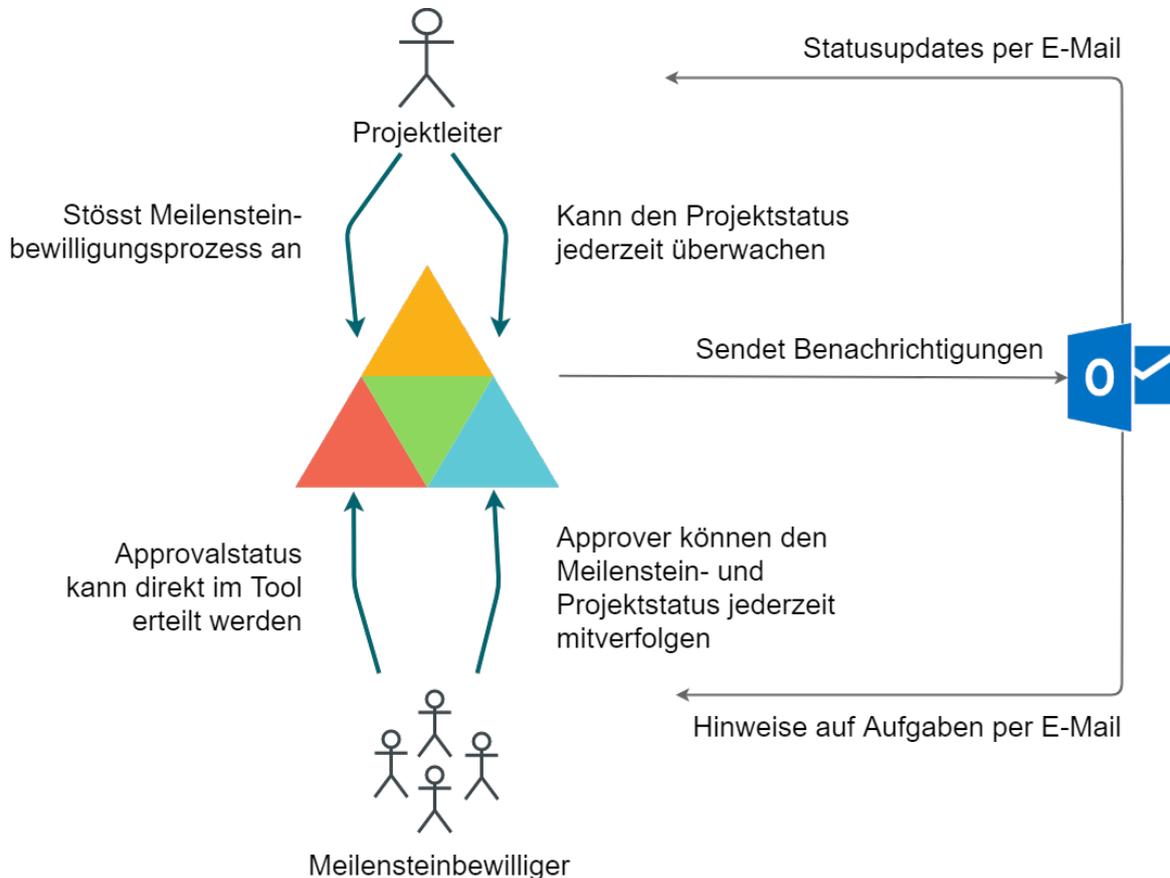


Abbildung 23. Der neue Bewilligungsprozess soll komplett in der neuen Lösung umgesetzt sein. Der Umweg über Excel entfällt, Email wird nur noch als Erinnerungsmedium genutzt.

## 3.3. Nichtfunktionale Anforderungen

Die funktionalen Anforderungen an die Applikation und insbesondere an den umzusetzenden Meilensteinbewilligungsprozess werden durch folgende, nichtfunktionale Anforderungen ergänzt.

### 3.3.1. User Experience & Design

Der Meilensteinbewilligungsprozess als Teil des VCR Ökosystems wird heute von mehreren hundert Anwendern genutzt. Je nach Aufgabe des Anwenders ist es realistisch, dass er diesen Ablauf jedoch nur selten nutzt, da Meilensteine oft Monate auseinanderliegen. Die Applikation soll für alle Anwender einfach zu bedienen sein, unabhängig davon, wie oft sie die Applikation nutzen und wie versiert Sie im Umgang mit der Applikation sind. Der Anwender soll möglichst wenig *falsch* machen können. Um dies zu erreichen, soll auf eine übersichtliche, einfache und durchgängige Gestaltung geachtet werden. Wo sinnvoll sollen Hilfselemente wie z.B. Tooltips dem Anwender kontextorientierte Hilfe anbieten. Die Applikation muss immer klar visualisieren und kommunizieren, welche Aktionen in der Benutzeroberfläche vorgenommen werden können und welche nicht.

### 3.3.2. Datenkonsistenz

Da die neue Applikation an das bestehende VCR System angebunden wird, muss die Datenkonsistenz in jedem Fall gewährleistet sein. Es sind geeignete Massnahmen zu implementieren, um den Datenaustausch zwischen den Systemen zu gewährleisten. Bis auf weiteres wird VCR als *Data Master* agieren. Dies bedeutet, dass die Stammdaten weiterhin bei VCR liegen und wir deren Konsistenz beim Abgleich mit unseren Daten gewährleisten müssen.

### 3.3.3. Performanz

Die Applikation soll durch den Anwender bedient werden können, ohne dass Interaktionen mit der Software zu signifikanten Wartezeiten führen. Generell wird die Performanz der heutigen VCR Lösung als ungenügend betrachtet, weswegen die neue Applikation bei vergleichbaren Operationen spürbar schneller sein muss.

### 3.3.4. Sicherheitsanforderungen

#### Authentisierung & Authentifizierung

In VCR gibt es heute eine Benutzerliste mit allen für die Benutzung von VCR zugelassenen Mitarbeitern. Damit wird sichergestellt, dass nur Benutzer, welche einmal einen *Access*

*Request* gestellt haben, überhaupt Zugriff auf die Applikation haben (sofern dieser genehmigt wurde). Bei Organisationsänderungen oder Mutationen kann diese grundsätzliche Zugriffsberechtigung auch wieder entzogen werden.

Die aktuelle VCR Benutzerliste muss während des Parallelbetriebs in die neue Applikation gespiegelt werden, damit sie weiterhin an einem zentralen Ort gepflegt werden kann.

Um in der neuen Applikation die Identität des aufrufenden Benutzers garantieren zu können, müssen diese schlussendlich im *Active Directory (AD)* überprüft werden. Es ist jedoch nicht mehr zeitgemäss, eine eigene Login-Maske zu implementieren, wo Benutzername und Passwort selbst abgefragt und im Hintergrund im Active Directory geprüft werden. Stattdessen soll die Applikation die Identifizierung und Überprüfung des Benutzers an einen *Identity Provider* delegieren.

### **Autorisierung und rollenbasierte Zugriffsberechtigungen**

Die Abläufe müssen so abgesichert sein, dass ausschliesslich Anwender mit entsprechenden Rollen Aktionen durchführen können. Da im ersten Schritt lediglich der Meilensteinbewilligungsprozess in der neuen Applikation umgesetzt wird, müssen im Rahmen dessen lediglich sichergestellt werden, dass das initiieren eines Meilensteinbewilligungsprozesses nur durch den jeweiligen Projektleiter und Projektqualitätsmanager möglich ist. Die Bewilligung oder Ablehnung eines Meilensteins soll zudem nur durch den jeweils angefragten Nutzer möglich sein.

### **Datenhoheit & Datensicherheit**

Die heutige VCR Applikation ist ausschliesslich aus dem Internet zugreifbar. Es muss sichergestellt werden, dass auch in Zukunft keinerlei Informationen über das Internet zugreifbar sind.

## 4. Software Architektur

### 4.1. Rahmenbedingungen

Als erstes soll kurz auf die Randbedingungen und Vorgaben, welche unsere Freiheiten bei der Entwicklung und Umsetzung einschränken, eingegangen werden.

#### 4.1.1. Technologiestack

Bei der Wahl des Technologiestack ist, wenn immer möglich, auf die bisher bereits beim Auftraggeber im Einsatz stehenden sowie die bereits erfolgreich im Prototyp eingesetzten Technologien zurückzugreifen. Genauere Erläuterungen zum Technologieeinsatz sind dem nachfolgenden Kapitel [Technologieentscheidungen](#) zu entnehmen.

#### 4.1.2. Infrastruktur & Deployment

Da Zugriffsmöglichkeiten auf die bestehende VCR Datenbank zwingend für die Umsetzung dieser Applikation sind, muss die neue Applikation Zugriff auf das Siemens Netzwerk haben. Diese Rahmenbedingung und der explizite Wunsch unseres Auftraggebers, Kontrolle über die Infrastruktur zu haben, führen dazu, dass unsere Applikation *On-Premise* bei Siemens gehostet werden muss.

## 4.2. Lösungsstrategie

Nachfolgend soll ein kurzer Überblick über die grundlegenden Entscheidungen und Lösungsansätze, die Entwurf und Implementierung des Systems prägen, aufgezeigt werden.

### 4.2.1. Architekturübersicht

Die bestehende Applikationsarchitektur wurde durch den Auftraggeber genehmigt, und soll den Bedürfnissen der Applikation angepasst werden. Die hier gezeigte Abbildung soll die gesamte Architektur aus der Vogelperspektive darstellen, um sich einen Überblick über die verschiedenen Komponenten und deren Vernetzung verschaffen zu können.

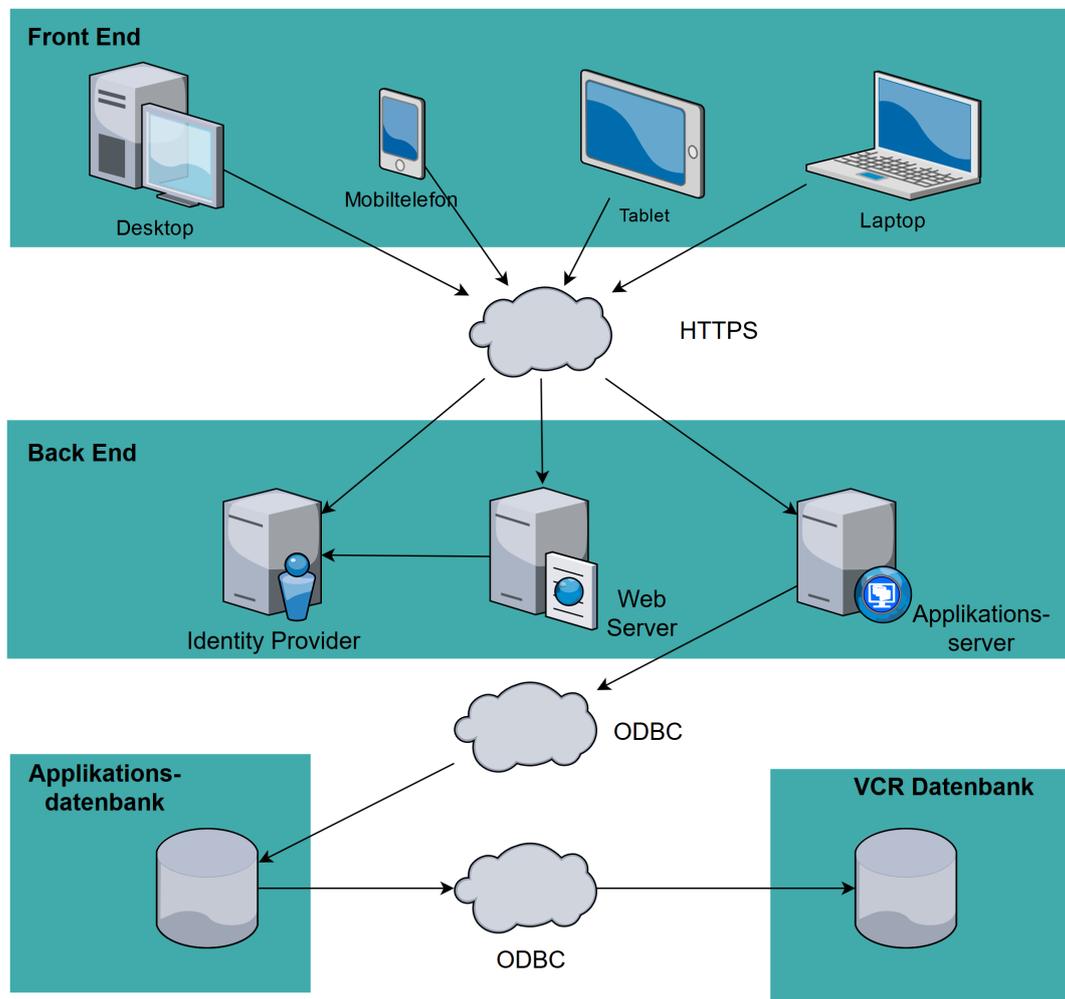


Abbildung 24. High Level Architekturübersicht der neuen Lösung

Detaillierte Ausführungen der einzelnen Schichten folgen in den Kapiteln [Frontend](#), [Backend](#) und [ETL Implementierung](#). Das physische Deployment wie es derzeit bei Siemens in Betrieb ist wird in [\[Integrationsumgebung bei Siemens\]](#) beschrieben.

## 4.2.2. Technologieentscheidungen

Die Auswahl der verwendeten Entwicklungstechnologien ist, mit wenigen Ergänzungen, seit der Studienarbeit unverändert. Die hier präsentierte Kurzfassung beruht auf der Technologieevaluation der Studienarbeit ([SA-7]), und erwähnt die wichtigsten, grundlegenden verwendeten Technologien. Technologien, welche während der Bachelorarbeit evaluiert und analysiert wurden, werden später ausführlicher behandelt (zum Beispiel das [Deployment](#) und das [Extract Transform Load](#) Prinzip).

### Frontend Technologien

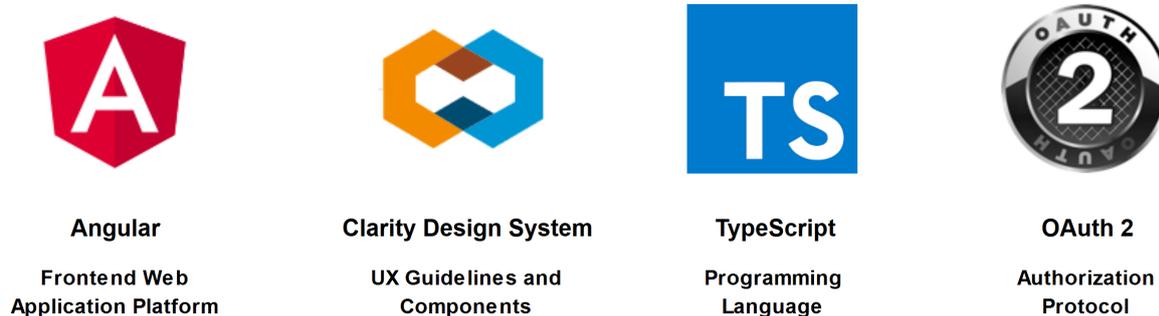
Der Frontend Technologiestack wurde in der *Studienarbeit* evaluiert. Die Wahl der *UI/UX Library* fiel auf das *Clarity Design System*. Als Frontend Framework wird auf *Angular* gesetzt:

*Die [evaluierten] Vorteile von Angular (die Struktur, der grössere Umfang, Built-in Unterstützung für MVC & DI) erachten wir als sehr wichtige Eckpfeiler einer Applikation. Die verfügbare Roadmap und ein Konzept für Long Term Support sorgen zudem für Planungssicherheit, was ein weiterer Vorteil [von Angular] ist.*

— [SA-3]

Die Entscheidung für und der Einsatz von *Clarity* und *Angular* haben sich bisher bewährt und werden deshalb als gegeben erachtet.

Somit ergibt sich der Einsatz folgender Frontend Technologien:



*Abbildung 25. Überblick über die wichtigsten eingesetzten Frontend Technologien*

## Backend Technologien

Bei den Backend Technologien ist eine Evolution des aktuellen Technologiestack [von VCR] gewünscht. Es sollen weiterhin .NET eingesetzt werden und die Datenbank soll weiterhin MS SQL basiert sein, da sich dieser Technologiestack zum einen in der Bestandlösung VCR als auch in anderen Projekten bewährt hat.

— [SA-2]

Diese Vorgaben galten bereits zu Beginn der [SA].

Bei der Wahl der zu verwendenden MS SQL Server Version gab es ebenfalls Einschränkungen. Die höchste Version, welche bei Siemens durch einen sogenannten *Security Measurement Plan* freigegeben ist und somit eingesetzt werden darf, ist die Version 2016. Neuere Versionen (2017 und 2019) dürfen noch nicht eingesetzt werden.

Daraus ergibt sich die Nutzung folgender Technologien im Backend:



Abbildung 26. Überblick über die wichtigsten eingesetzten Backend Technologien

## Deployment Plattform

Beim Erstellen des Prototyps für die Studienarbeit war noch keine endgültige Deploymentstrategie für eine eventuelle Produktivversion erarbeitet worden. Die Vorgaben der Studienarbeit lauteten:

Der Betrieb der Infrastruktur soll auch in Zukunft ökonomisch sein, dies soll u.a. durch eine möglichst homogene Infrastruktur erreicht werden. Wenn die Lösung wieder auf Windows Servern läuft [...] wäre dies sehr vorteilhaft für unsere Kosteneffizienz. Darunter soll jedoch nicht die Lösung leiden.

— [SA-4]

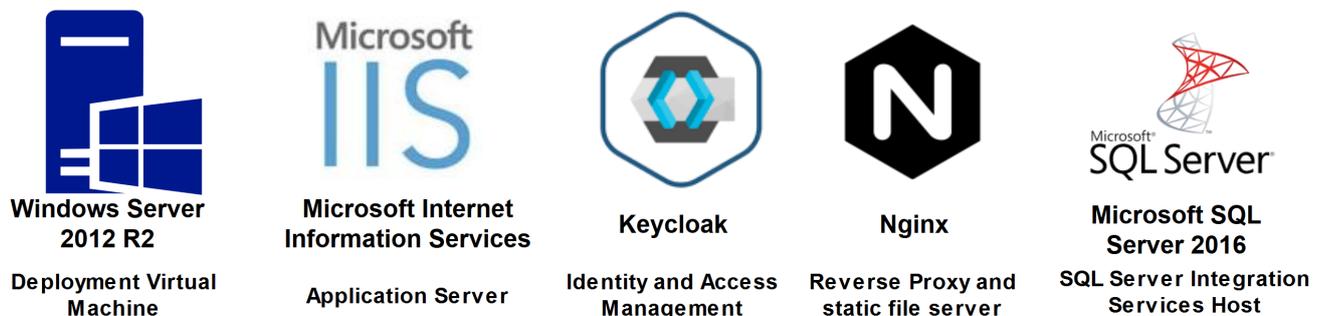
In der Folge wurde eine möglichst flexible, Docker basierte, Deployment Strategie entwickelt, welche vor allem darauf fokussiert war, während der Entwicklung schnell iterieren zu können.

Entwicklungs- und Staging Umgebung waren dabei sehr ähnlich und haben eine produktionsnahe Umgebung auf den Entwicklungsrechnern ermöglicht.

Für die endgültige Inbetriebnahme der Software bei Siemens wurde von unserem Auftraggeber entschieden, dass Betrieb der neuen Version ähnlich wie bisher gestaltet werden muss. Somit sollen sämtliche Dienste auf Windows Server 2012 R2 Virtual Machines gehostet werden.

Gleichzeitig soll die Möglichkeit einer cloudbasierten Deploymentstrategie offengehalten werden, falls diese in Zukunft als Alternative zum Windows basierten *On-Premise* Hosting genutzt werden soll. Bei Siemens werden bereits verschiedene Cloud-Technologien für Deployments eingesetzt, darunter OpenShift und Amazon Web Services. Da zur Zeit dieser Bachelorarbeit keine zusätzlichen Cloudanforderungen bekannt sind, ist Docker als allgemeine Containerplattform eine gute Wahl, und bietet eine gute Basis für eine zukünftige Migration zu einem der zahlreichen Cloudanbieter.

Für das Deployment der entwickelten Applikation innerhalb von Siemens wurde eine Windows Server 2012 R2 Instanz bereitgestellt. Die Applikation selbst, wie auch sämtliche benötigten Dienste wurden darauf installiert und sind in laufendem Betrieb. Daraus ergeben sich folgende bei Siemens für das Hosting verwendete Technologien:



*Abbildung 27. Aktuelle Deployment Plattform bei Siemens*

## Entwicklungsumgebung

Die während der Studienarbeit als Deploymentplattform verwendete Umgebung kann nicht als Produktivumgebung weiterverwendet werden. Sie wurde jedoch als Stagingumgebung weiterverwendet, um zu testen, ob lauffähige Deployment Artefakte erstellt wurden. Auch die bereits aufgesetzten Entwicklungstools wurden weiterverwendet, wie beispielsweise Sonar für die Überprüfung der Code Qualität oder Portainer zur Überwachung der Staging Umgebung. Auch auf den lokalen Entwicklungsmaschinen wurde weiterhin mit den gleichen Technologien gearbeitet, da sich diese als praktisch erwiesen haben.

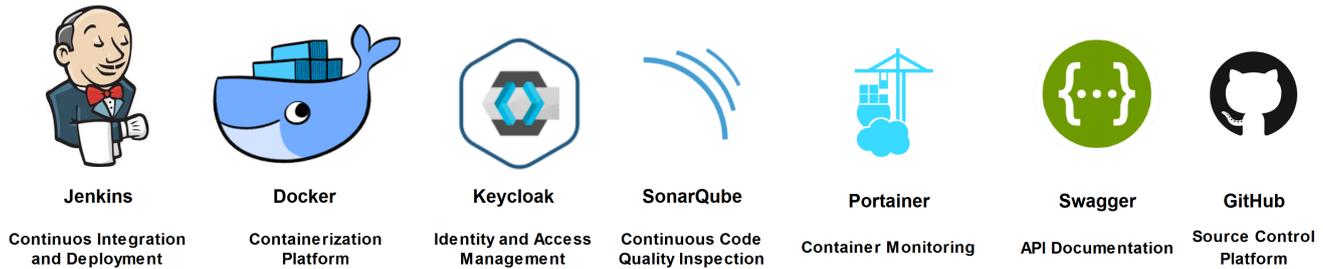


Abbildung 28. Überblick der verwendeten Entwicklungs Tools

## Extract Transform Load

Um den Parallelbetrieb beider Applikationen (VCR und das neue Cockpit) mit gleichem Datenstamm zu gewährleisten, wurde entschieden, die benötigten Daten per ETL Job zu synchronisieren. In einem ersten Schritt sollen dazu Daten aus der bestehenden Lösung unidirektional in die neu entworfene Datenbankstruktur übertragen werden. Auf diese Weise soll gezeigt werden, wie solche Umwandlungen gestaltet werden können, um eines Tages eine komplette Migration der bestehenden Datenbank durchführen zu können. Von Siemens wurde zu diesem Zweck ein SQL Benutzer erstellt, welcher ausschliesslich lesend Zugriff auf die bestehende Datenbank hat. Dies hat uns ermöglicht, sowohl die bestehenden Daten wie auch deren Struktur zu evaluieren, um daraus ETL Jobs zur Datensynchronisation gestalten zu können.

Es wurde entschieden, für die Umsetzung dieser ETL Jobs das von Microsoft entwickelte *SQL Server Integration Services* Softwarepaket zu nutzen. Dies, nachdem wir eine ausführliche Evaluation durchgeführt haben, welche im Anhang unter [ETL Software Evaluation](#) nachgelesen werden kann.

## Externe Schnittstellen

Bei der Evaluation der bestehenden Datenbank und den benötigten Daten der neuen Applikation wurde klar, dass nicht alle Daten direkt aus der Datenbank abgegriffen werden können. Gewisse Daten werden zur Laufzeit durch die Business Logik von VCR erst aufbereitet, damit sie für den aktuellen Meilensteinprozess verwendet werden zu können. Eine vertiefte Analyse der VCR Meilensteinschnittstelle hat ergeben, dass es sinnvoll wäre, die für den neuen Meilensteinprozess zu verwendenden Daten über eine externe Schnittstelle zu integrieren.

In Rücksprache mit dem Projektleiter von VCR wurde eine XML Schnittstelle für Meilensteindaten in Auftrag gegeben. Diese Schnittstelle bietet per SOAP ein XML Dokument an, welches die gleichen Daten enthält wie ein von VCR generiertes Excel Sheet für die Meilensteinbewilligung. Die detaillierte Analyse der so entstandenen XML Dokumente befindet sich im Anhang unter [\[Schnittstellenanalyse\]](#).

## 4.3. Bausteinsicht

### 4.3.1. Frontend

Das in Browsern lauffähige Angular Frontend basiert auf den Empfehlungen aus der Angular Architektur Übersicht [\[Angular-Arch\]](#). Das Diagramm zeigt die Abhängigkeiten der umgesetzten Single Page Application von links nach rechts.

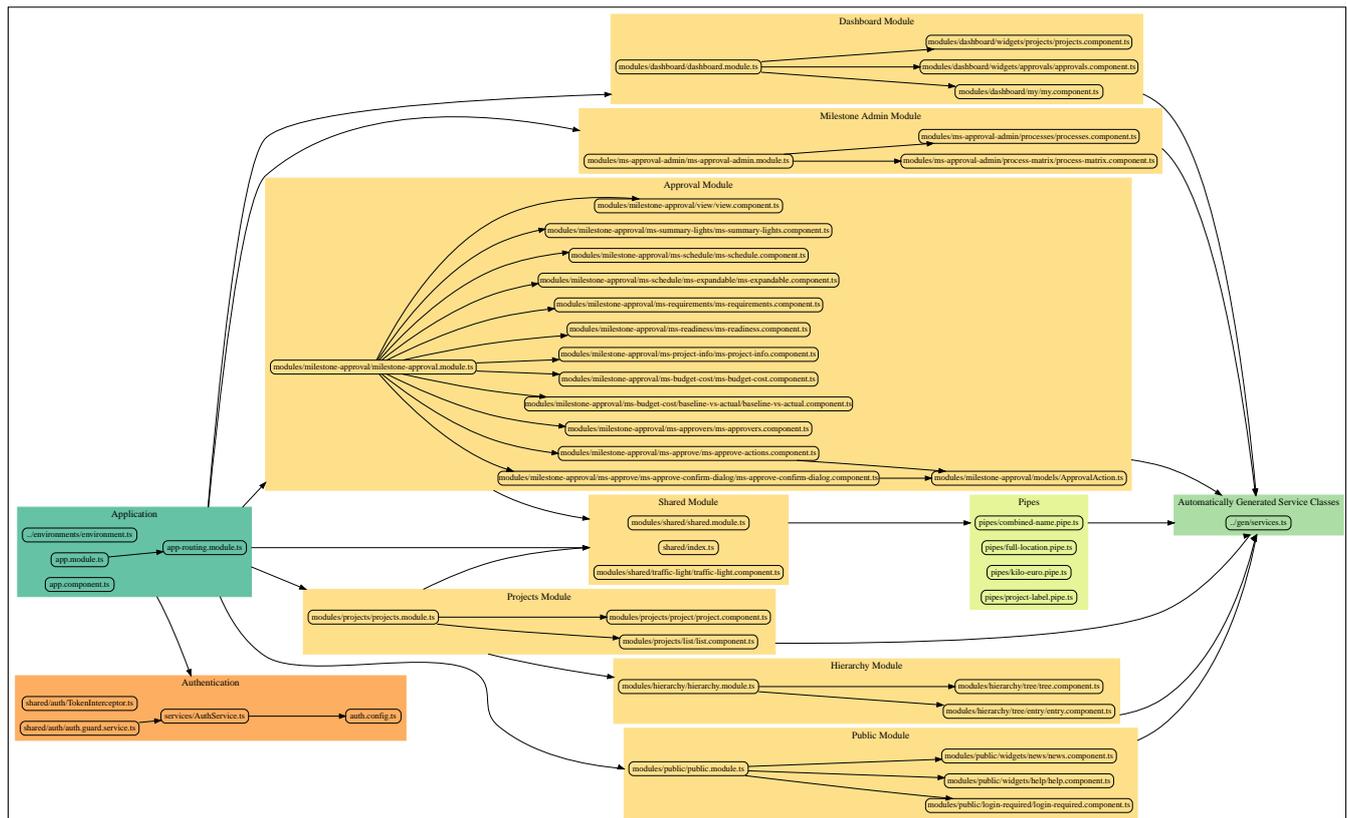


Abbildung 29. Paketübersicht von Angular Frontend



Bild aus Source Code erstellt mit [\[Madgel\]](#) und Layout manuell angepasst.

*Application* ist das Einstiegsmodul welches auch das Routing aufsetzt. *Authentication* ist eine applikationsübergreifende Komponente welche für den OpenDI Connect Flow zuständig ist und auch die [\[JWT\]](#) (JSON Web Token) Silent Refresh Vorgänge übernimmt. Alle orange eingefärbten Module sind für die Userinteraktion zuständig, wobei Module zusammen mit den ihnen zugehörigen Komponenten dargestellt werden. *Pipes* dienen der Umformatierung von Daten für die usergerechte Darstellung. Alle Module nutzen einen oder mehrere Services aus dem von Swagger generierten API Client. Aus technischen Gründen befinden sich alle Serviceklassen in einer einzigen Datei. Es gibt jedoch für jeden API Endpoint eine separate Serviceklasse. Die benötigten Services werden per Dependency Injection den verwendenden Komponenten zur Verfügung gestellt.

### 4.3.2. Backend

Das Backend ist eine REST API, implementiert mit ASP.NET Core ([\[NET-CORE-API\]](#)). Der Aufbau entspricht einer klassischen mehrschichtigen Architektur. Bei der Umsetzung wurden die allgemeinen Richtlinien von Microsoft zur API Entwicklung mit Dotnet Core befolgt ([\[ASPNET-Tutorial\]](#)). Die Data Transfer Objects (DTOs) werden verwendet, um die Lebenszeit der von Entity Framework genutzten Verbindung zum SQL Server so kurz wie möglich offen zu halten. Gleichzeitig können für Nutzer der API irrelevante Daten weglassen werden, ohne dies explizit konfigurieren zu müssen.

Da es sich beim Frontend um eine Single Page Application (SPA) handelt, ist die Backend Architektur bewusst einfach gehalten. Um dem Nutzer im Browser ein flüssiges Bedienerlebnis zu ermöglichen, werden viele Daten bereits verknüpft und aggregiert an die SPA geschickt. Dies hat zur Folge das weniger, dafür aber umfangreichere Netzwerkabfragen durchgeführt führen.

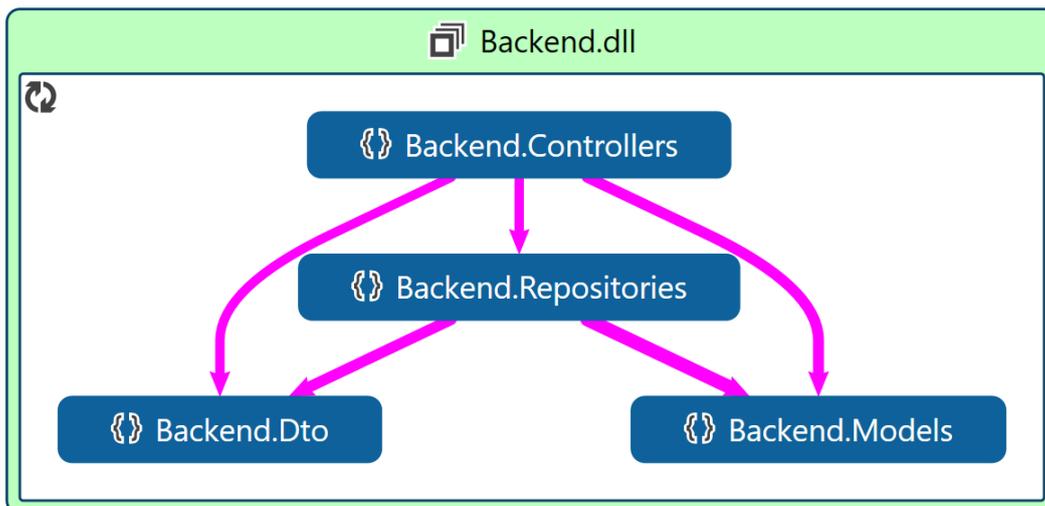


Abbildung 30. Paketübersicht der Backend Architektur

## Kurzes Code Review

Zu sehen ist, dass alle Dependencies nur von oben nach unten gehen, und dass die *Controller* über die *Repository* Klassen auf den *Data Access Layer* zugreifen. In der aktuellen Lösung hat jeder *Controller* ein eigenes *Repository*, über welches er Daten abfragen kann.

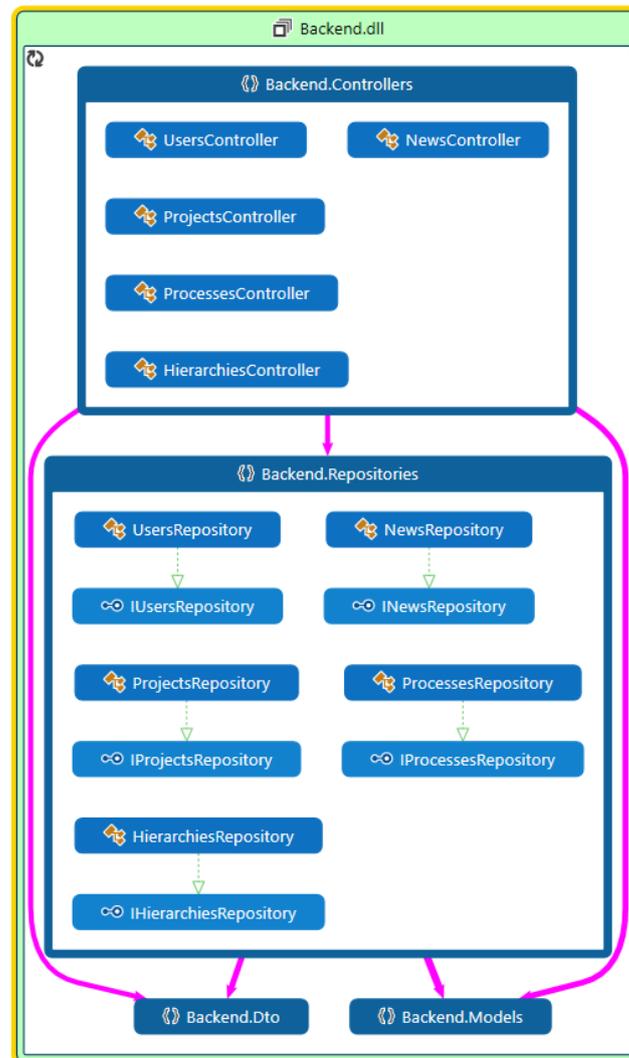


Abbildung 31. Detailübersicht Controller und zugehörige Repositories

Wie in den folgenden Screenshots zu sehen ist, greifen alle *Controller* mit Ausnahme des `ProjectsController` nur auf *Repositories* zu und arbeiten ausschliesslich mit Data Transfer Objects. Für den `ProjectsController` könnte dies mit einem noch ausstehenden *Refactoring* ebenfalls mit wenig Aufwand umgesetzt werden. Die Controller sind dabei nur für die eigentliche Darstellung der Daten zuständig, während die Geschäftslogik in Bezug auf Datenbankabfragen in den *Repositories* gekapselt ist.

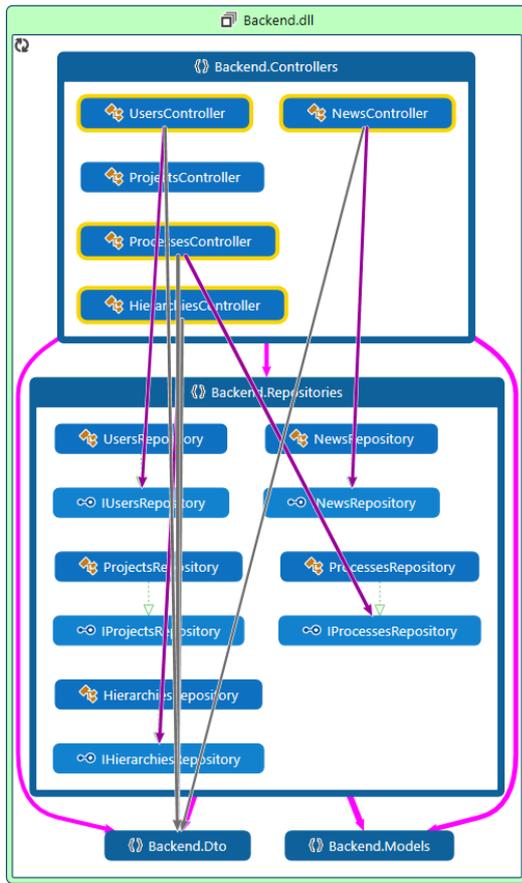


Abbildung 32. Abhängigkeiten der Controller

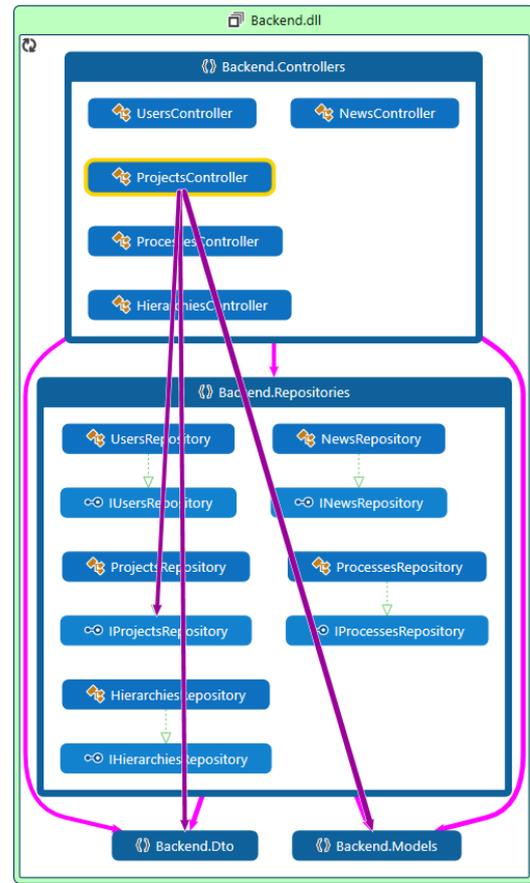


Abbildung 33. Kapselung 'ProjectsController'

Hier sieht man, wie der `ProjectsController` als einziger Controller die Datenkapselung bricht und direkt eine Dependency auf `Backend.Models` hat. Wenn Backend so überarbeitet würde, dass alle Controller nur noch über Repositories auf Daten zugreifen kann, liesse sich das Dependency Diagramm noch weiter reduzieren. So wären alle Controller ausschliesslich für die Behandlung von HTTP Abfragen zuständig, was zu folgender Gesamtübersicht führen würde.

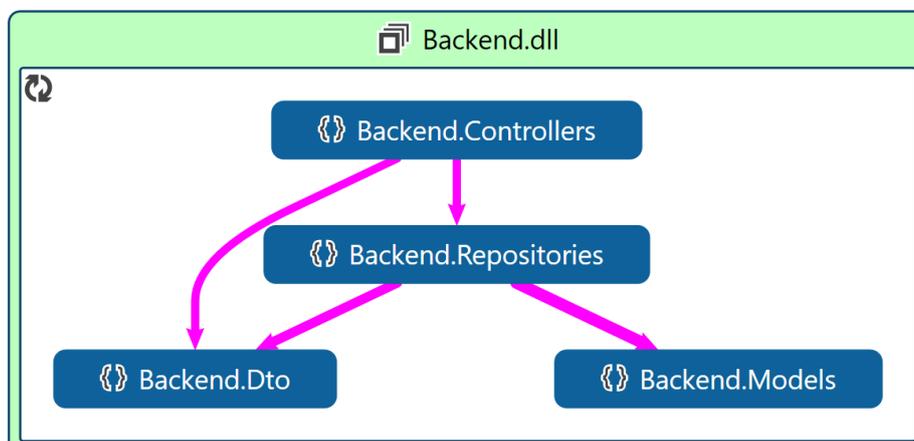


Abbildung 34. Angestrebte Zielarchitektur nach Vollendung des Refactorings

## Komplexität

Die simple Implementierung der Controller- und Repository Namespaces steht in Kontrast zur Komplexität der DTOs und Models. Wie bereits erwähnt werden die Daten vom Backend bereits verknüpft in geschachtelten Payloads an die Single Page Application geschickt. Das hat den Vorteil, dass die Anzahl Abfragen reduziert werden können, welche notwendig sind, um bei Interaktionen in der Benutzeroberfläche neue Daten anzuzeigen. Der Nachteil dieser Umsetzung in Backend Architektur äussert sich jedoch darin, dass viele Datenbankverbindungen auch in den DTOs wieder abgebildet werden müssen. Hier hat sich AutoMapper als praktisch erwiesen, da er sich implizit um diese Umwandlung aus geschachtelten Model Objekten zu Data Transfer Objekten kümmert oder diese mit minimalem Aufwand umgesetzt werden können. Auf diese Art lässt sich die hier gezeigte Komplexität auf einen Namespace beschränken, ohne die Schranken zwischen den Schichten durchbrechen zu müssen.

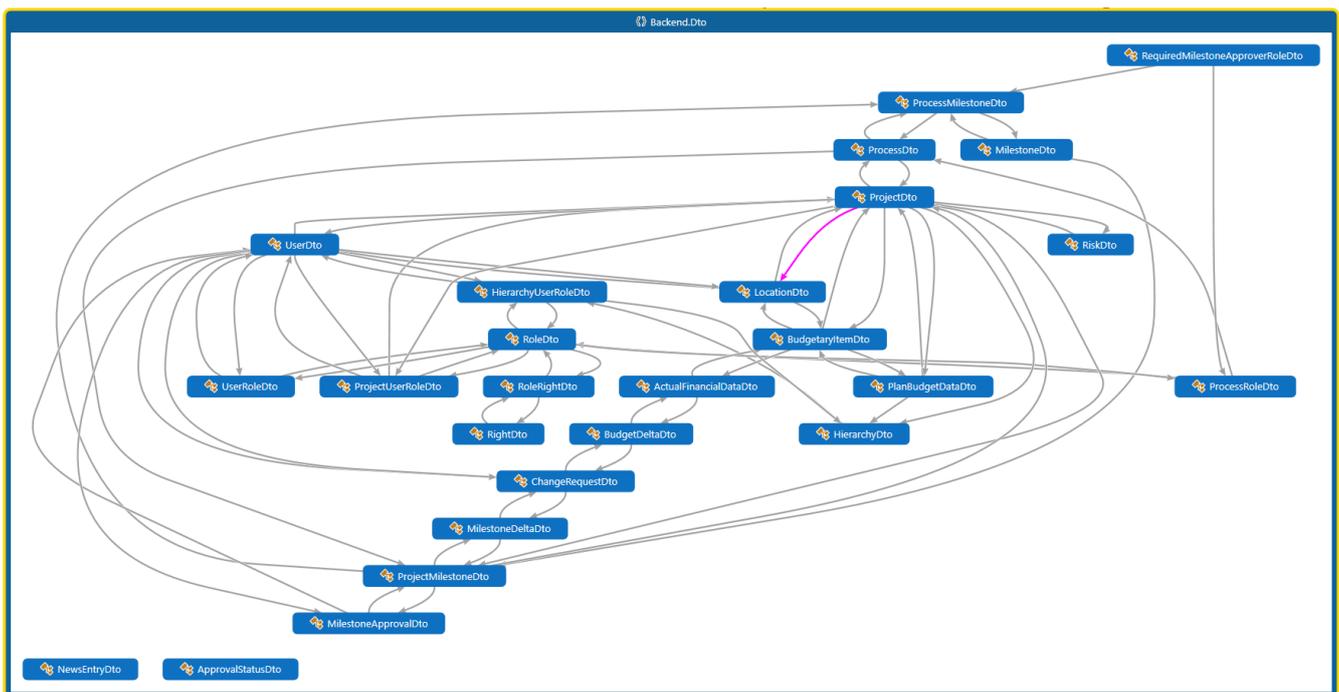


Abbildung 35. Hohe zyklomatische Komplexität in der DTO Schicht



Bilder wurden aus dem Source Code erstellt mit Hilfe der Studentenlizenz von Visual Studio Enterprise.

## API

Die *Controller*-Annotationen und Funktionsdeklarationen definieren den Aufbau der API. Aus ihnen werden die *Swagger* Definitionen generiert.

### *Beispiel einer Controller-Klasse*

```
1 [Route("api/projects")]
2 [ApiController]
3 public class ProjectsController : ControllerBase {}
```

### *Beispiel einer Controller-Funktion*

```
1 [HttpGet("{cid}", Name = "GetProject")]
2 public ActionResult<ProjectDto> GetById(string cid)
```

Aus der Kombination dieser Annotationen ergibt sich die HTTP REST API. Die API kann beispielsweise folgendermassen aufgerufen werden:

### *Beispielaufruf eines fiktiven Projekts (verkürzt)*

```
curl -X GET \  
  https://sinv-56083.edu.hsr.ch/projects/1200031-ZUG \  
  -H 'Accept: application/json'
```

## Models

Die Klassenstruktur der *Model*-Klassen bestimmt gleichzeitig auch die Struktur der Datenbank. Eine *Model*-Klasse entspricht einer Tabelle der Datenbank, und die Schlüsselrelationen werden anhand der Collections der *Model*-Klasse bestimmt. Entity Framework Core [\[EF-2\]](#) generiert alle resultierenden SQL Migrationsskripte. Die resultierenden Tabellenbeschreibungen können auch für die Generierung von *Mock*-Daten verwendet werden. Das folgende Beispiel zeigt eine C# Klassenbeschreibung einer Entity Framework *Model*-Klasse, und einen Ausschnitt der entstandenen Datenbank mit den entsprechenden Fremdschlüsseln.

### Beispiel einer Entity Framework Model-Klasse

```

1 public class Process
2 {
3     public int Id { get; set; }
4     public string Name { get; set; }
5     public string Description { get; set; }
6     public ICollection<ProcessMilestone> ProcessMilestones { get; set; }
7     public ICollection<ProcessRole> ProcessRoles { get; set; }
8     public ICollection<Project> Projects { get; set; }
9     public int? VcrOldId { get; set; }
10 }

```

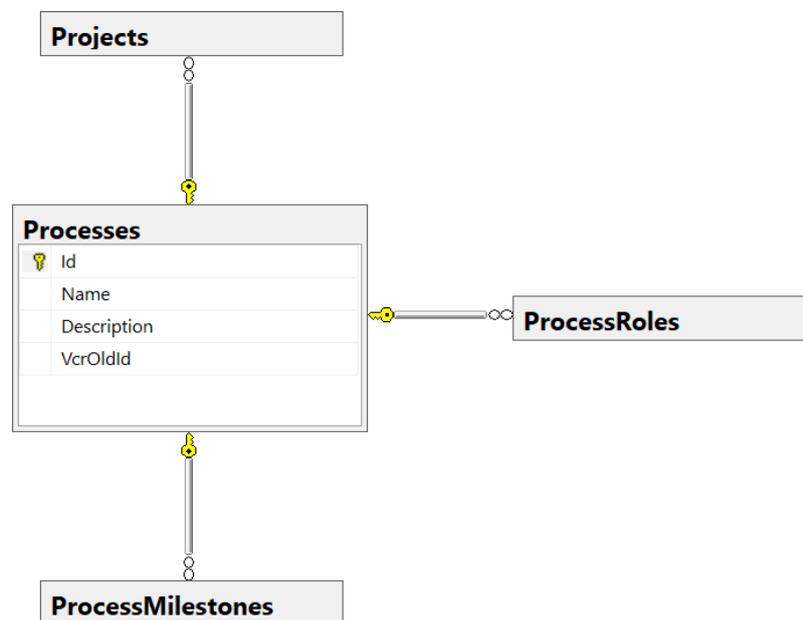


Abbildung 36. Resultierende Datenbanktabelle mit den durch Fremdschlüsselbeziehungen verbundenen Tabellen.

### 4.3.3. Datenmodell

Während der [SA] wurde ein Datenmodell entwickelt, welches die Datenstruktur der Applikation beschreibt. Da es sich um ein relationales Modell handelt und wir auf ein objektrelationales Mapping setzen, findet sich diese Struktur praktisch eins zu eins in der Datenbank wieder.

Die Daten im nachfolgenden Datenmodell lassen sich in vier Gruppen unterteilen:





Diagramm erstellt mit [\[ERD\]](#) als Teil der AsciiDoc Dokumentgenerierung.

#### 4.3.4. Extract, Transfer, Load

Als Teil der neuen Architektur wurde ein neues Datenbankmodell entworfen, welches verschiedene Verbesserungen der Datenhaltung ermöglichen soll. Um den Parallelbetrieb sowohl der alten wie auch der neuen Lösung mit gleichen Stammdaten zu ermöglichen, müssen die Daten aus VCR für die neue Applikation erreichbar gemacht werden.

Eine Lösung für dieses Problem wäre gewesen, aus dem Backend auf beide Datenbanken gleichzeitig zuzugreifen. Dies hätte jedoch dazu geführt, dass viele sehr ähnliche Abfragen hätten geschrieben werden müssen, um alle Datenabgleiche programmatisch durchzuführen. Dieses Vorgehen ist nicht zielführend, wenn es darum geht, eine Perspektive für eine komplette Ablösung der alten Plattform zu ermöglichen. Das Ziel war es also, ohne eine eigentliche Abhängigkeit auf die alte VCR Datenbank zu haben, die neu modellierte Datenbank von Anfang an bereits voll Nutzen zu können. Dieser Ansatz hat ferner den Vorteil, dass das neue Datenbankdesign direkt auf seine Tauglichkeit geprüft werden kann, da es von Anfang an aktiv genutzt wird und Schwachstellen so bereits während der Entwicklung aufgedeckt werden und nicht etwa erst bei der Inbetriebnahme.

#### ETL Implementierung

Um all diese Aspekte zu berücksichtigen, wurde entschieden, mit *ETL (Extract, Transfer, Load) Tools* zu arbeiten ([Extract Transform Load](#)). Es wird SQL Server Integration Services ([ETL Software Evaluation](#)) verwendet, um Daten aus der VCR Datenbank abzugreifen, diese an die neuen Tabellen anzupassen und dann in die neue Datenbank zu schreiben. In der aktuellen Lösung werden die notwendigen Daten für den überarbeiteten Meilensteinbewilligungsprozess in regelmässigen Abständen von der produktiven VCR Datenbank per ETL Job in die neue Datenbank übertragen. Der folgende Screenshot zeigt die abstrakte Ansicht des gesamten ETL Workflows.

Ein detailliertes Beispiel eines einzelnen Ablaufs ist der Laufzeitsicht im Kapitel [SSIS Datenfluss](#) zu entnehmen.

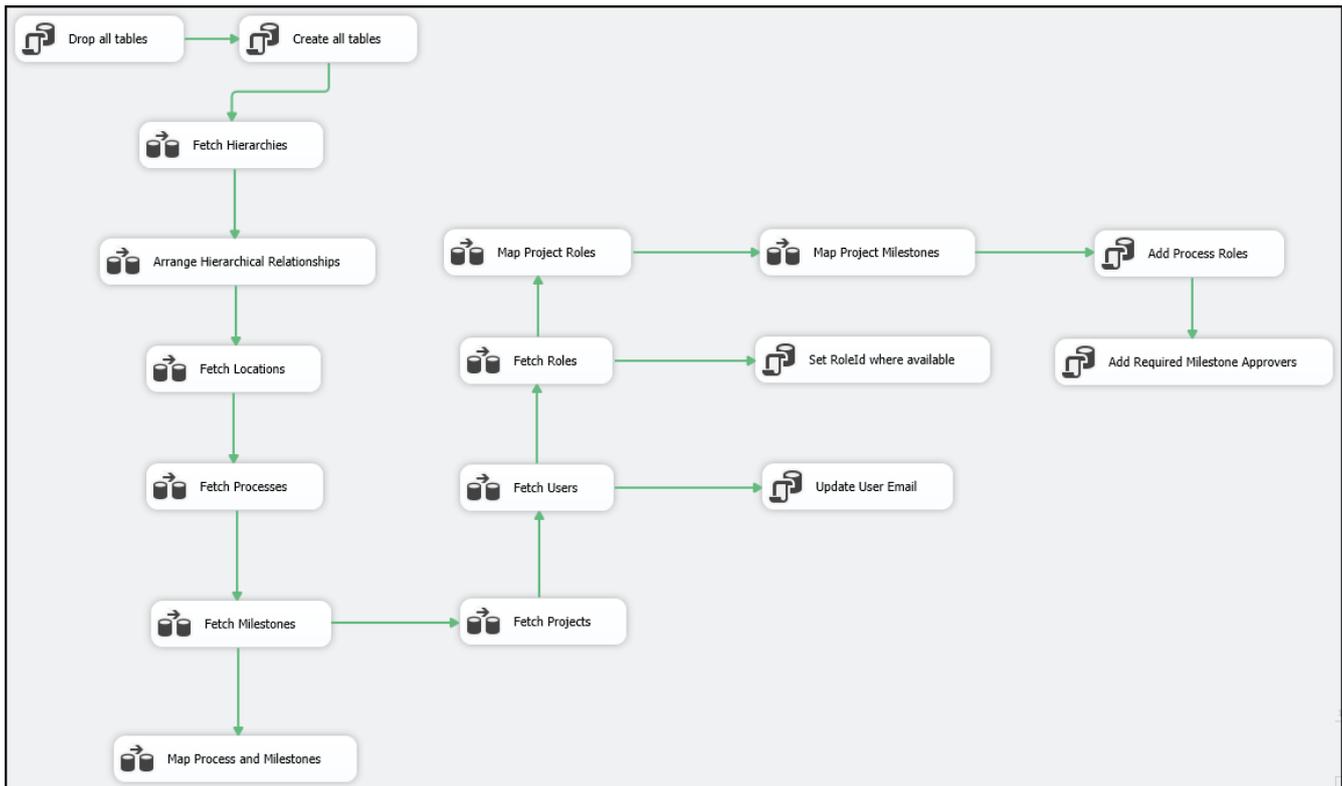


Abbildung 38. Screenshot der Datensynchronisationskomponente in Visual Studio

Der Datenfluss des ETL Workflows beginnt oben links mit der Erstellung neuer, leerer Tabellen. Diese werden dann sukzessive mit den benötigten Daten aufgefüllt. Die Reihenfolge der transformierten Daten wird durch die Fremdschlüsselabhängigkeiten der involvierten Tabellen bestimmt. Die *Foreign Key Constraints* werden absichtlich nicht deaktiviert. Einerseits, um konsistente Daten am Ende der Transformation garantieren zu können, andererseits aber auch um bestehende Dateninkonsistenzen der bestehenden Lösung aufzudecken. Mit diesem Vorgehen wurden beispielsweise duplizierte Hierarchien in der bestehenden VCR Datenbank gefunden und dem heutigen Betreiber mitgeteilt.

Durch die Datenanalyse, welche notwendig war um die ETL Jobs zu bauen, haben wir zusätzlich zu den erlangten Daten auch detaillierten Einblick in die bestehende Datenbank erlangt. Ein tieferes Verständnis der bestehenden Lösung hilft dabei, bereits begangene Fehler zu vermeiden und Unzulänglichkeiten auszubessern. Auf Basis der heutigen Daten ist es zum Beispiel schwierig zu erfahren, ob ein Projekt überhaupt aktiv ist oder nicht und ob dieses korrekt abgeschlossen wurde. Aus diesen Erkenntnissen lassen sich auch für die Zukunft neue Schlüsse ziehen, zum Beispiel wie in Zukunft eine einheitlich Definition eines Projektabschlusses definiert wird.

## Anpassungen an der Datenbank

Während der Implementierung der SSIS Workflows musste eine Lösung für das Verbinden von

alten und neuen Daten gefunden werden. Da die Datenbanken mit Ausnahme der ETL Jobs keine eigentlichen Verbindungen haben, können Joins nicht einfach arbiträr über verschiedene Tabellen vorgenommen werden. Es musste also eine Möglichkeit gefunden werden, um Daten in der neuen Datenbank mit ihrer Herkunft in der alten Datenbank in Verbindung zu bringen. Eine Lösung wäre gewesen, die gleichen IDs anstatt der automatisch generierten zu verwenden. Es können jedoch nicht alle Entitäten eins zu eins miteinander in Verbindung gebracht werden. Auch wird mit dieser implizierten Art der Verbindung nicht offensichtlich, welche Tabellen mit ihren Vorgängern in Verbindung stehen und welche nicht.

Aus diesen Gründen wurde entschieden, als pragmatische Lösung die alten IDs in separaten Spalten in den neuen Datenbanktabellen mitzuführen. Diese Methode hat auch den Vorteil, dass die Spalten in Zukunft gelöscht werden können, wenn keine Notwendigkeit für ETL Jobs mehr bestehen, ohne die Struktur der neuen Datenbank zu beeinträchtigen. Da keine *Constraints* auf diese Spalten angewendet werden, kann der Löschvorgang mit einer unkomplizierten SQL Migration vorgenommen werden.

*Beispiel einer Tabelle mit alter ID in den Zeilen 5 & 6*

```
1 CREATE TABLE [Hierarchies] (  
2   [Id] int NOT NULL IDENTITY,  
3   [LevelName] nvarchar(max) NOT NULL,  
4   [CombinedName] nvarchar(max) NOT NULL,  
5   [ParentId] int NULL,  
6   [VcrOldId] int NULL,  
7   [VcrOldParent] int NULL,  
8   CONSTRAINT [PK_Hierarchies] PRIMARY KEY ([Id]),  
9   CONSTRAINT [FK_Hierarchies_Hierarchies_ParentId]  
10  FOREIGN KEY ([ParentId])  
11  REFERENCES [Hierarchies] ([Id])  
12  ON DELETE NO ACTION  
13 );
```

## Performanz des Datenaustauschs

Die Verwendung von ETL soll einen raschen Datenaustausch ermöglichen, um Modifikationen zeitnah in der neuen Datenbank verfügbar zu machen. Ziel der ETL Implementierung war also nicht nur die Ermöglichung des Datenflusses, sondern auch diesen schnell und daher in kurzen Abständen ausführen zu können. Nach anfänglich langsamen Durchläufen konnte mit relativ wenig Aufwand eine Durchlaufzeit von unter einer Minute für die gesamte ETL Pipeline erreicht werden.

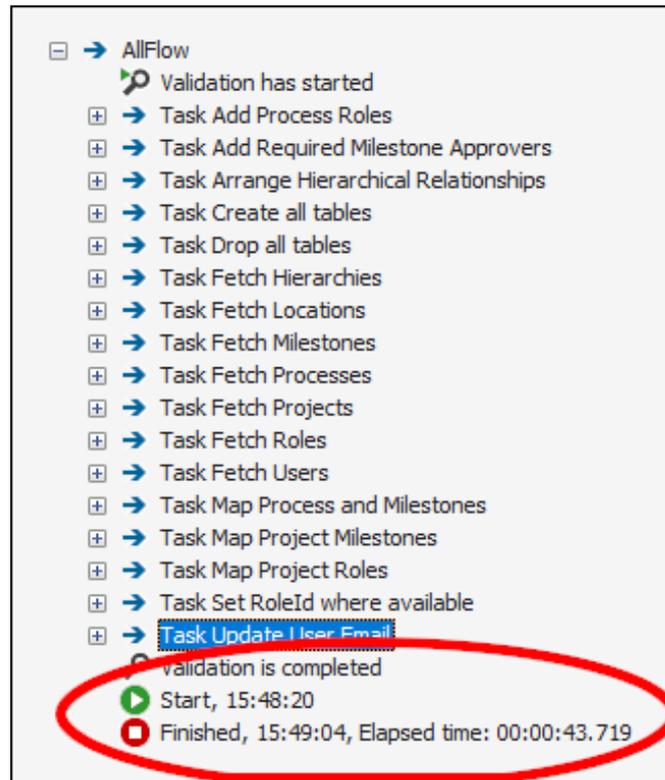


Abbildung 39. Laufzeitübersicht eines kompletten SSIS Durchgangs

Am meisten Einfluss auf die Laufzeit der ETL Jobs hatten zwei Faktoren. SSIS ist zeilenorientiert und transformiert in der Standardkonfiguration Zeile für Zeile. Dies führt zu einer tiefen Auslastung der Datenbanktransaktionen und somit sehr vielen überflüssigen Datenbankverbindungen. *Caching* und *Bulk Insertion* werden jedoch von SSIS unterstützt und können gezielt konfiguriert werden, um die Leistung zu steigern. So werden viele Zeilen auf einmal abgefragt, transformiert, und wiederum viele andere Zeilen auf einmal geschrieben.

In einem zweiten Schritt wurden diejenigen Vorgänge, welche nach der ersten Optimierung noch langsam waren, genauer analysiert und wo möglich durch handgeschriebenes SQL ersetzt. Die grafisch entworfenen SSIS Workflows sind verständlich, jedoch manchmal zu wenig flexibel, um optimierte Abläufe abbilden zu können. Wo nötig können grafisch erstellte Komponenten durch SQL Scripts ersetzt werden, um Performanzprobleme gezielter adressieren zu können. Als Beispiel für einen auf diese Weise optimierten Vorgang dient erneut die *Hierarchies* Tabelle aus dem vorhergehenden Beispiel. Das Abbilden der alten Hierarchie und ihren Fremdschlüsseln in der neuen Tabelle hat im grafisch entwickelten Workflow lange gedauert, weil viele *Row Look-ups* gemacht werden mussten, diese aber nicht explizit optimiert werden konnten. Stattdessen wurde die Hierarchie in zwei Schritten neu erstellt. Zuerst wird die Hierarchie im Schritt "Fetch Hierarchies" komplett geschrieben, ohne die *FK\_Hierarchies\_Hierarchies\_ParentId* Referenz zu setzen. In einem zweiten Schritt werden mit einem kurzen SQL Script und der Verwendung der *VcrOldId* und *VcrOldParent*

aus der alten Tabelle die ursprünglichen Verbindungen wiederhergestellt.

#### *Wiederherstellung der Hierarchie durch ein SQL Skript*

```
1 UPDATE child
2 SET ParentId = parent.Id
3 FROM Hierarchies child
4 INNER JOIN Hierarchies parent on
5   child.vcroidparent = parent.vcroidid
```

Bei vielen potentiellen Optimierungen muss jedoch auch abgewogen werden. Einige manuell optimierte SQL Abfragen können eine erhebliche Komplexität aufweisen, was die Wartung solcher Komponenten erschwert. Als Beispiel ist im Anhang unter [SQL Code Ausschnitt](#) die optimierte Anfrage, welche um nur die neuesten Einträge des jeweiligen Projektmeilensteins in die neue Datenbank überträgt, exemplarisch aufgeführt. Diese Abfrage hat den Prozess zwar beschleunigt, wäre bei Änderungen der abgefragten Tabellen aber schwieriger zu warten als ein vergleichbarer Dataflow.

### Integration mit externen Schnittstellen

SSIS ermöglicht die Anbindung zahlreicher Schnittstellen und erlaubt dabei den Datenaustausch zwischen Datenbanken ([SQL Server Integration Services](#)). Beispielsweise können Daten auch von APIs im Internet abgegriffen und per ETL Job in eine Datenbank gespeichert werden. Diese Flexibilität ermöglicht es, auf die in [Externe Schnittstellen](#) besprochene SOAP Schnittstelle per ETL Job zuzugreifen.

Es war geplant, alle für die Meilensteinbewilligung benötigten Daten auf diese Weise abzugreifen, und in der neuen "*milestone declaration data*" Tabelle abzulegen ([Datenmodell](#)). Ziel war es, für jeden Projektmeilenstein die dazugehörigen Meilensteinbewilligungsdaten jederzeit aus der Datenbank abrufen zu können, ohne direkt auf die Businesslogik von VCR zuzugreifen. Diese Vorgehensweise soll es ermöglichen, alle Daten für die Meilensteinschnittstelle bereitzustellen und auf diese Weise eine endgültige Migration weg vom aktuell benutzten *Excel Sheet* zu ermöglichen. Leider sind während der Implementierung einige Probleme aufgetreten, welche eine Umsetzung dieses Ziels im Rahmen der Bachelorarbeit verunmöglicht haben. Besonders die Komplexität einer massgeschneiderten API Anbindung und die aktuelle Performance der XML Schnittstelle haben dazu geführt, dass dieses Feature nicht mehr während der Bachelorarbeit umgesetzt werden konnte.

### Komplexe Implementierung

Die bereits erwähnte Flexibilität von SSIS ermöglicht es zwar technisch, verschiedenen

Schnittstellen in den ETL Workflow einzubinden. Es ist jedoch abhängig von der spezifischen Schnittstelle, wie gut sich diese auf einen SSIS Workflow anpassen lässt. Am geeignetsten sind Daten, die ohne weitere Aufbereitung bereits in einzelnen Datenreihen vorliegen, da die Datenverarbeitung in SSIS auf die Bearbeitung von reihenorientierten Daten optimiert ist. Weil die VCR XML Schnittstelle jedoch als unvorbereitetes XML Dokument mit einer Ansammlung verschiedener Daten implementiert wurde, ist es notwendig, eine solche Datenverarbeitungs Komponente zu entwickeln. Es hat sich gezeigt, dass einer solchen innerhalb der Bachelorarbeit zeitlich leider nicht mehr drin liegt.

### Unzureichende Antwortzeit der XML Schnittstelle

In der Einführung wurde bereits die unzureichende Antwortzeit von VCR erwähnt:

Die Schlechte Performance der Applikation äussert sich z.B. in sehr langen Ladezeiten bei diversen Vorgängen.

— Anforderungsanalyse

Bei der Implementierung der SSIS Komponente, um auf die VCR Schnittstelle zuzugreifen hat sich ergeben, dass die durchschnittliche Zugriffszeit pro Meilenstein etwa 15 Sekunden beträgt. Daraus ergibt sich für eine einzige Abfrage aller Projektmeilensteine bereits eine Laufzeit von mehreren Stunden. Dies ist in Anbetracht der allgemeinen optimierten Laufzeit der ETL Pipeline inakzeptabel. Zur Zeit der Verfassung dieser Arbeit wurde für dieses Problem noch keine zufriedenstellende Lösung gefunden.

## 4.4. Laufzeitsicht

### 4.4.1. SSIS Datenfluss

Der im Kapitel [ETL Implementierung](#) vorgestellte Workflow besteht aus vielen "Data Flow" genannten Komponenten, welche für die Datensammlung und Transformation zuständig sind. Hier soll einer dieser Data Flows vorgestellt werden, um einen Einblick in die Komponenten der ETL Jobs zu geben.

Es wird der Map Process and Milestones Data Flow beschrieben, welcher einem Prozess seine per Definition zugehörigen Meilensteine zuteilt (siehe [Aktuell bei \[Siemens\] im Einsatz stehende Geschäftsprozessstypen. Die vier mittleren Projekttypen haben eine variierende Anzahl Meilensteine. Die beiden Prozessstypen Yearly und Agile Release Train wiederum kennen keine Meilensteine.](#)). Es ist empfehlenswert, jeden Data Flow möglichst unabhängig von anderen zu implementieren, da dies Tests vereinfacht und die Kopplung zwischen verschiedenen Data Flows minimiert. Dank dem Einsatz von Caching führen wiederholte Abfragen auch auf grossen Tabellen kaum zu Performanzeinbussen.

Zuerst verschaffen wir uns jedoch einen Überblick über das grafische Layout des Data Flows. Der Datenfluss ist von oben nach unten gerichtet. In einem vorherigen Schritt wurden zudem die Prozess- und Meilensteindaten bereits importiert, so dass diese Daten benutzt werden können.

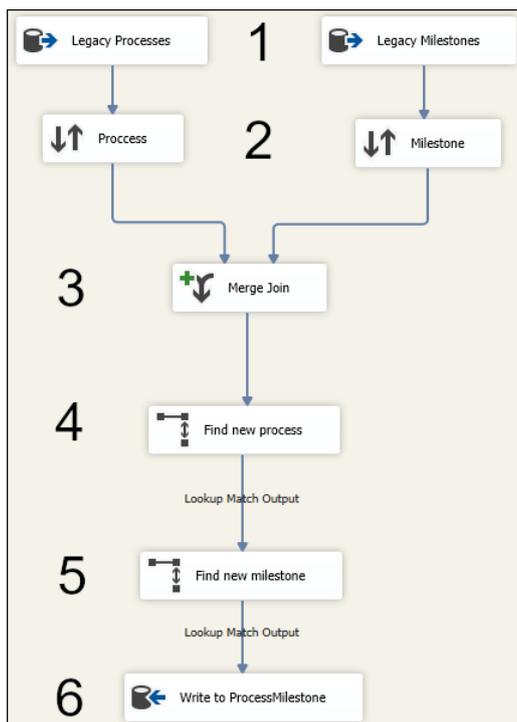


Abbildung 40. Data Flow für Zuweisung von Prozessmeilensteinen

1. Prozesse und Meilensteine werden aus der alten Datenbank geholt.
2. Beide Tabellen werden nach Id sortiert.
3. Die Daten der alten Tabelle werden über ihren Fremdschlüssel mit Inner Join verbunden.
4. Der Eintrag in der neuen Prozesstabelle wird anhand seiner `vcrOldId` gesucht und der zuvor in Merge Join erstellten Tabelle als weitere Spalte hinzugefügt.
5. Der Eintrag in der neuen Meilensteintabelle wird anhand seiner `vcrOldId` gesucht und der zuvor in Find new process bereits erweiterten Tabelle wiederum hinzugefügt.
6. Die so zusammengeführten Prozessmeilensteine werden in ihre Tabelle geschrieben.

Der Inner Join der Daten in **Schritt 3** resultiert in einer Tabelle, welche aus den Spalten beider eingehenden Tabellen besteht. Wo nötig, müssen doppelt verwendete Namen (in diesem Fall `id`) mit einem Alias versehen werden, um diese eindeutig identifizieren zu können.

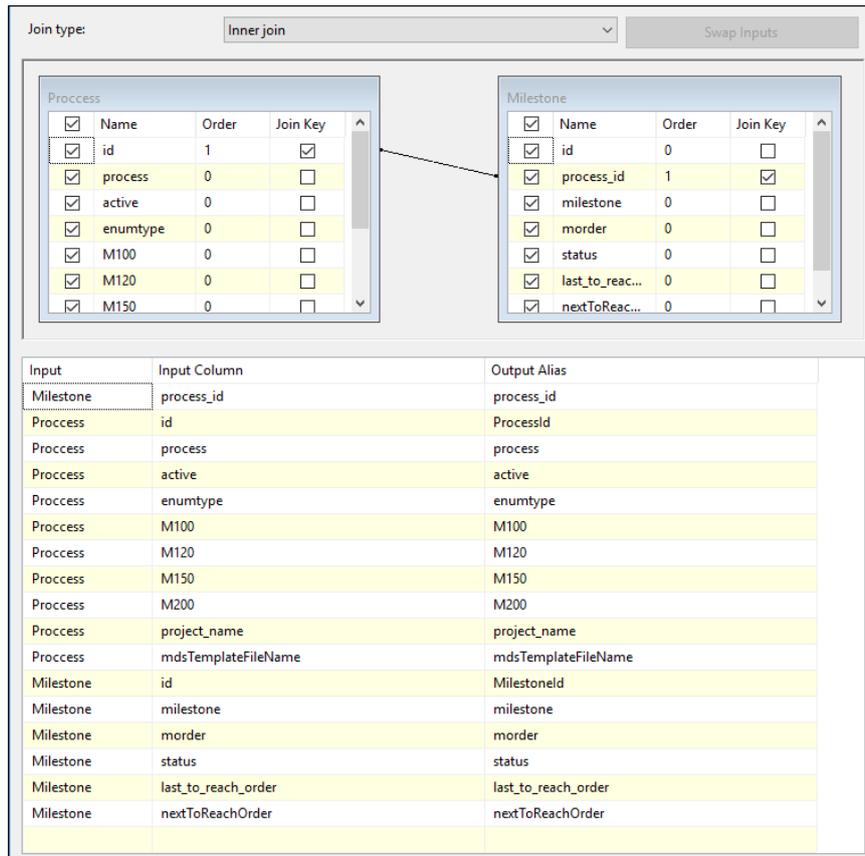


Abbildung 41. Tabelle als Resultat des Joins

In **Schritt 4 & 5** werden mit Hilfe der `VcrOldId` der jeweiligen Tabellen die dazugehörigen Einträge gesucht. Dazu wird die Beziehung dieser beiden Spalten über das UI "Create Relationships" hergestellt.

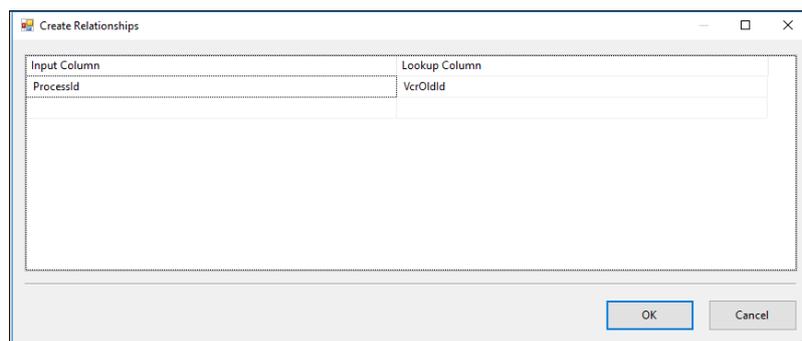


Abbildung 42. Suchen von Eintrag in neuer Prozesstabelle

Wenn die Verbindung hergestellt wurde, können die `_Lookup` Columns der neuen Tabelle der alten als zusätzliche Spalten hinzugefügt werden. Dies verbindet die alten mit den neuen Prozess- und Meilensteindaten. Da über den Join in Schritt 3 bereits die Beziehung der alten Daten hergestellt wurde, stimmen diese nun mit den neu zu erstellenden Beziehungen überein.

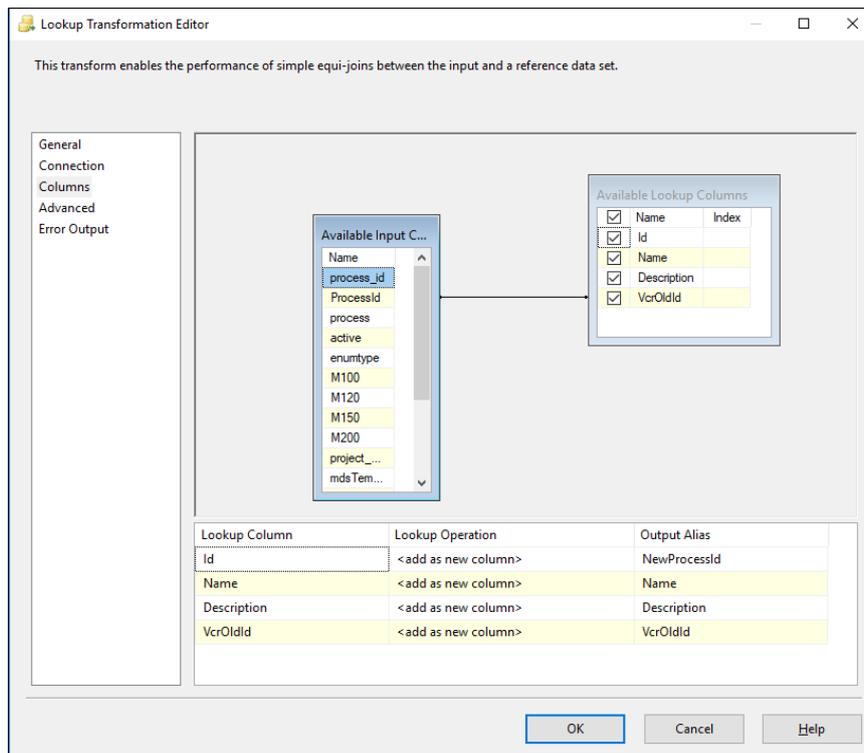


Abbildung 43. Hinzufügen von zusätzlichen Spalten

Im sechsten und letzten Schritt können nun die Daten in die neue Prozessmeilensteintabelle übernommen werden. Gleichzeitig lassen sich alle in der neuen Tabelle nicht benötigte Daten wegfiltern, da nur die Werte der gewünschten Spalten übernommen werden können.

#### 4.4.2. SSIS Datenkonsistenz

Neben dem Transformieren von Daten kann SSIS auch dazu verwendet werden, die Daten währenddessen zu analysieren. Daten, welche zum Beispiel in einer *Lookup* Komponente nicht zugewiesen werden, können über den *Error Output* geloggt werden. Auf diese Art lassen sich Dateninkonsistenzen bereits beim Transformieren der Daten erkennen, und es kann verhindert werden, dass Dateninkonsistenzen in die neue Datenbank propagiert werden.

Mit diesem Vorgehensweisen wurden zum Beispiel 13 Projekte der alten VCR Datenbank gefunden, welche Meilensteine deklarieren, die nicht dem verwendeten Prozess entsprechen (*auch hier wurden diese Erkenntnisse wider dem Betreiber von VCR gemeldet*). In der neuen Datenbank ist es deshalb nicht möglich, diesen ungültigen Fall überhaupt abzubilden. Demzufolge werden diese falsch erstellten Meilensteine nicht übertragen. Es liegt nun an den Entwicklern von VCR, diesen Misstand in den bestehenden Daten zu beheben.

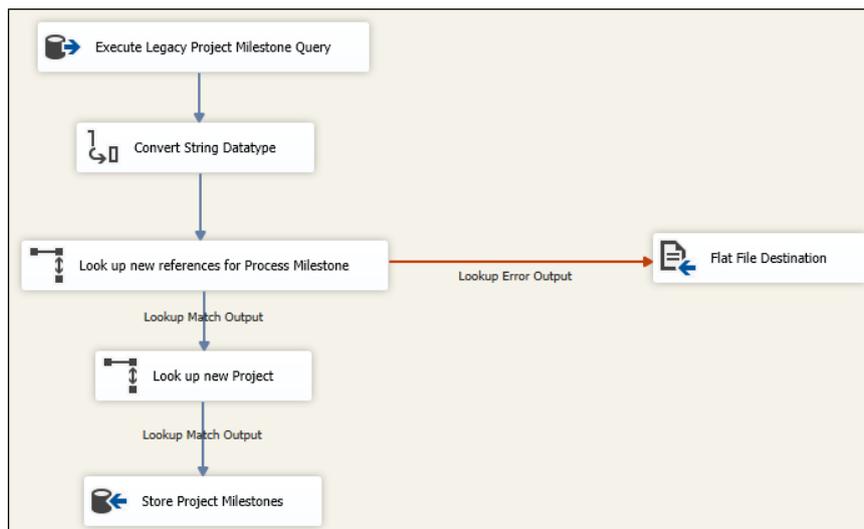


Abbildung 44. Beispiel von aufgetretenen Fehlern in einem Data Flow

*Beispiel eines Error Logs von Projekten mit falschen Meilensteinen*

```

1 455, 1723, 3639, 12, M100,3,PEP      ,2008-03-31,True ,Reached,Reached
2 7717,41879,238777, 1,FG0, 2,PEP2005,2007-10-01,True ,Reached,Reached
3 7723,42539,241346,12,M100,5,Pre-Dev,2007-04-30,False,Planned,Planned
4 7724,42541,241356,12,M100,5,Pre-Dev,2007-04-30,False,Planned,Planned
5 7729,44115,251421,12,M100,5,Pre-Dev,2007-10-01,True ,Reached,Reached
6 7729,41738,236715,12,M100,5,Pre-Dev,2007-10-01,True ,Reached,Reached
7 7729,41600,236569,12,M100,5,Pre-Dev,2007-10-01,False,Planned,Planned
8 7790,42045,239466,12,M100,5,Pre-Dev,2007-12-17,True ,Reached,Reached
9 7839,42531,241306,12,M100,5,Pre-Dev,2007-04-30,False,Planned,Planned
10 7840,42533,241316,12,M100,5,Pre-Dev,2007-04-30,False,Planned,Planned
11 7841,42537,241336,12,M100,5,Pre-Dev,2007-04-30,False,Planned,Planned
12 7854,42906,242901,12,M100,3,PEP      ,2007-12-15,False,Planned,Planned
13 7865,42535,241326,12,M100,5,Pre-Dev,2007-04-30,False,Planned,Planned
  
```

Es ist den aktuellen Betreibern bekannt, dass VCR teilweise mangelhafte Datenkonsistenz aufweist. Bei einer fortschreitenden Migration können weitere Fehler mit Hilfe von SSIS aufgedeckt und schrittweise korrigiert werden.

### 4.4.3. Authentication & Authorization Frontend

Nun wenden wir uns einigen Aspekten der Laufzeitsicht im Frontend zu.

Das Frontend nutzt zur Authentisierung und Autorisierung den OpenID Connect Implicit Flow ([\[oidc-flow\]](#)) basierend auf OAuth 2 ([\[oauth2\]](#)). Der Implicit Flow wird besonders von Browser Clients verwendet ([\[oidc-flow\]](#)).

Verwendet wird dazu die von OpenID zertifizierte Angular Library [\[angular-oauth2-oidc\]](#), welche unter anderem Silent Refresh ([\[silent-refresh\]](#)) unterstützt. Dies um zu verhindern, dass sich Benutzer periodisch neu einloggen müssen, sobald der Lebenszeit eines *Tokens* kurz vor dem Ablauf steht. Da derzeit bei Siemens noch keine Anbindung zu einem Identity Provider möglich ist (siehe [Rahmenbedingungen](#)), fungiert Keycloak sowohl als Zugriffskontrolle wie auch als Identity Provider. Aus Sicht der Applikation ergibt sich folgendes Sequenzdiagramm.

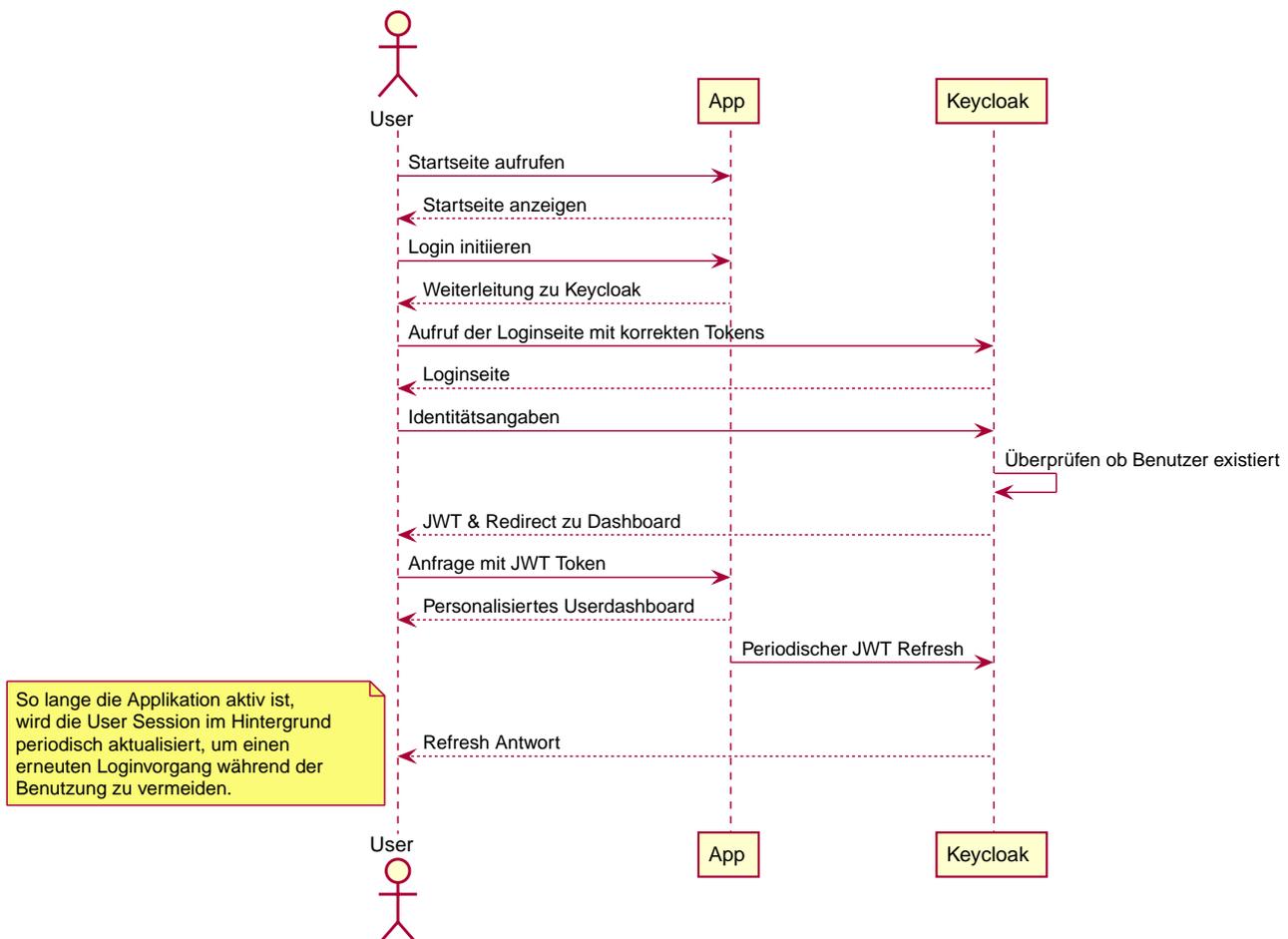


Abbildung 45. OIDC Implicit Flow aus Sicht der Applikation

#### 4.4.4. Authentication & Authorization Backend

Im Backend werden keine Usersessions verwaltet. Das Backend ist somit *Stateless* in Bezug auf Session Management. Dies würde u.a. die horizontale Skalierung der Applikation vereinfachen, da nicht verfolgt werden muss, bei welchem Backend eine User Session bereits angelegt wurde. Auch wenn dieser Anspruch aktuell nicht existiert, wird durch die aktuelle Umsetzung die Applikationslogik vereinfacht, da auf das Session Management im Backend verzichtet werden kann. Alle Abfragen an die API müssen mit einem Json Web Token ([JWT](#)) ausgestattet sein, welches von Keycloak im Hintergrund auf seine Gültigkeit überprüft wird. Ungültige Requests werden mit 401 - Unauthorized beantwortet.

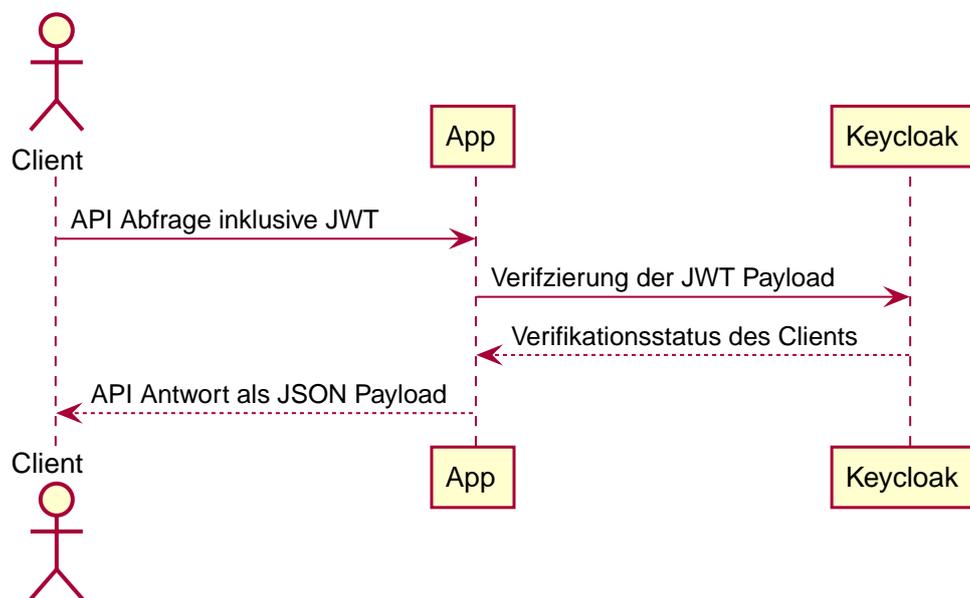


Abbildung 46. JWT Verifizierung im Backend



Dieses Bild und der Abschnitt entsprechen inhaltlich dem Kapitel 3.5.3. "Authentication & Authorization Backend" [\[SA-35\]](#) aus der Studienarbeit. Der Text wurde aufgrund schlechter Formulierung angepasst, um die Lesbarkeit zu erhöhen.

## 4.5. Verteilungssicht

Dieses Kapitel beschreibt die technische Infrastruktur und deren Abläufe.

### 4.5.1. Continuous Integration und Delivery

#### Continuous Integration

Jeder Push auf den Master Branch des Github Repositories startet einen vollautomatischen Build und Deployment Zyklus auf dem Entwicklungsserver. Nur wenn der komplette Build erfolgreich durchlaufen werden konnte, wird ein neues Deployment vorgenommen.

— [SA-5]

Die Continuous Integration Lösung, welche während der Bachelorarbeit verwendet wurde, unterscheidet sich kaum von der bereits bei der Studienarbeit verwendeten Infrastruktur. Die HSR hat uns ermöglicht, den gleichen Server für beide Arbeiten zu verwenden, womit nur minimale Anpassungen der Staging infrastruktur nötig waren.

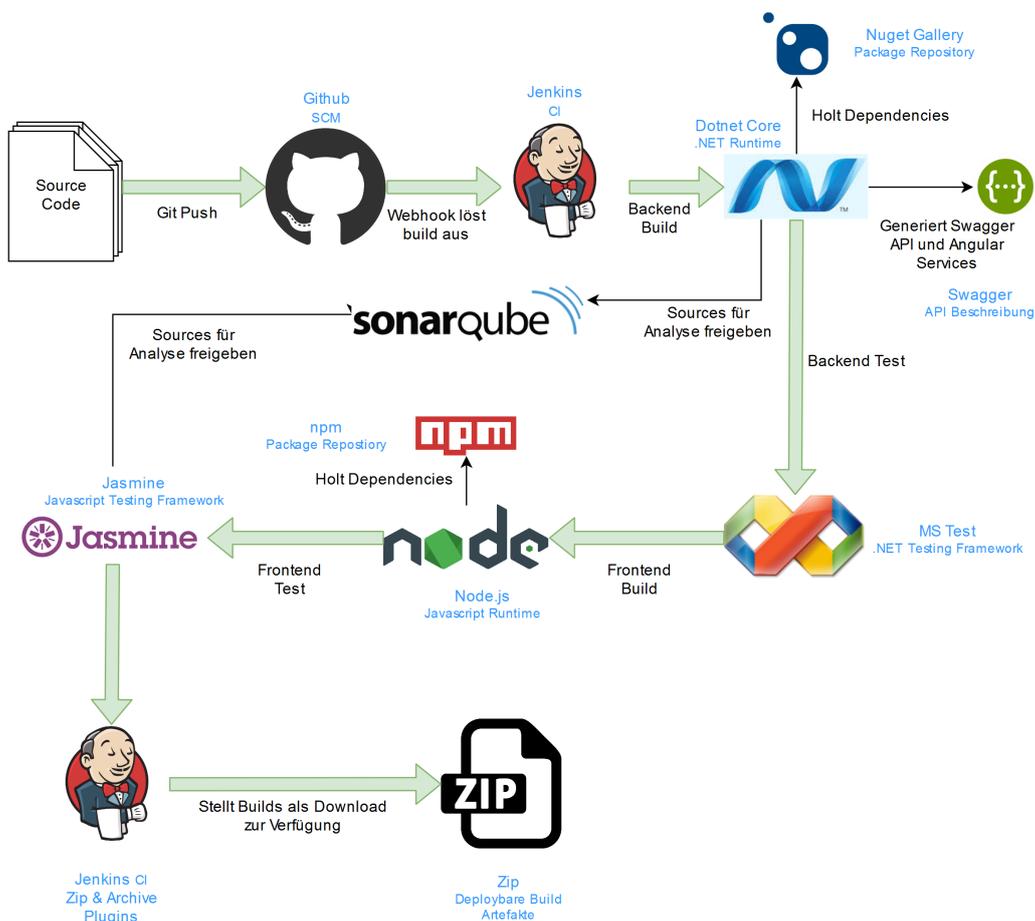


Abbildung 47. Continuous Integration Pipeline für das HSR Staging System

Als einzige Anpassung wurde der zuvor als eine einzige Jenkins Pipeline modellierte CI/CD Prozess in separate Vorgänge aufgeteilt. Die durch die CI Pipeline erstellten Artefakte werden in einer ZIP Datei zusammengefasst. Diese kann direkt vom Deployment genutzt werden. Das erfolgreiche Abschliessen dieser Pipeline stösst wiederum automatisch die Continuous Delivery Pipeline für das Staging System an. Somit unterscheiden sich die beiden Vorgänge ausschliesslich in der zusätzlichen Abspeicherung der Artefakte, damit diese per API zugänglich gemacht werden.

## Continuous Delivery Staging

Wie bisher wird jeder erfolgreiche Build des Master Branches direkt auf Staging bereitgestellt, mit dem einzigen Unterschied, dass dieser Schritt in einer separaten Pipeline abläuft. Falls ein Build nicht erfolgreich ist, bleibt das Staging System unangetastet.

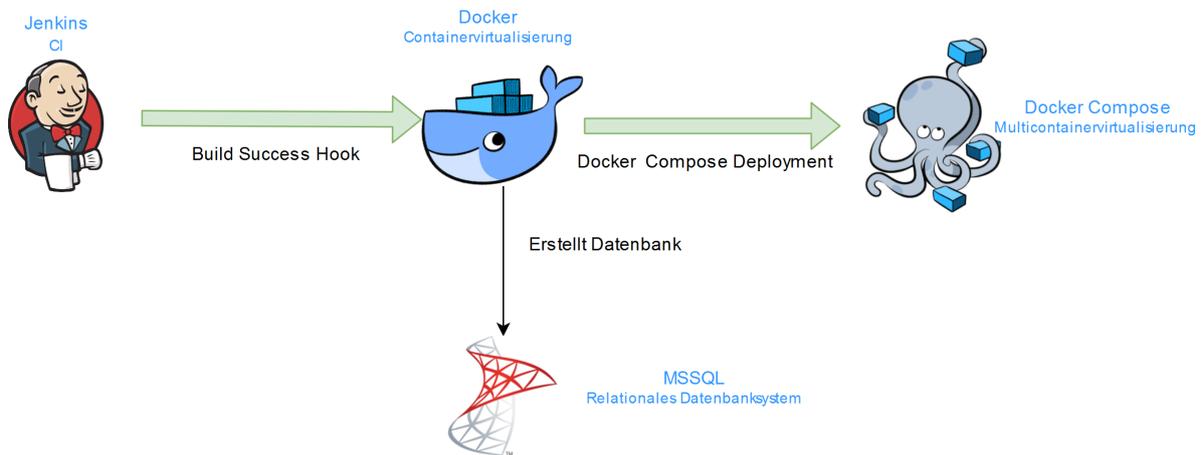


Abbildung 48. Deployment Pipeline für das HSR Staging System

### 4.5.2. Deployment

Das Deployment der ausgearbeiteten Lösung ist sehr variabel. Die ausgewählten Technologien sind systemunabhängig und bieten verschiedene Arten der Auslieferung. Die hier gezeigten Diagramme stellen die während der Arbeit verwendeten Konfigurationen dar, sind jedoch nicht die einzige Art, wie die Architektur aufgebaut werden kann. Ein *Reverse Proxy* dient in allen Fällen als Einstiegspunkt, um die eigentliche Architektur für den Client transparent zu halten. Neben der einfacheren Konfiguration bietet dies auch der Möglichkeit, in Zukunft auftretende Skalierungsprobleme zu adressieren, ohne Veränderungen am Client vorzunehmen, zum Beispiel durch Verwendung von *Load Balancing* in *Nginx* ([load-balancing](#)) oder Einsatz eines beliebigen anderen *Reverse Proxies* mit äquivalenter Funktionalität.

Neben den Vorteilen bei der Konfiguration unterstützt diese Art des Deployments zusätzlich die Isolation der verschiedenen Komponenten. Alle einzelnen Programme

kommunizieren nur über APIs miteinander und sind dadurch unabhängig der jeweiligen Implementierung.

Als gutes Beispiel der Flexibilität der Architektur dient SSIS. Dieser Dienst konnte ohne gezielte Änderungen an der bestehenden Architektur eingeführt werden und hat sich nahtlos in die Umgebung integriert.

### Staging

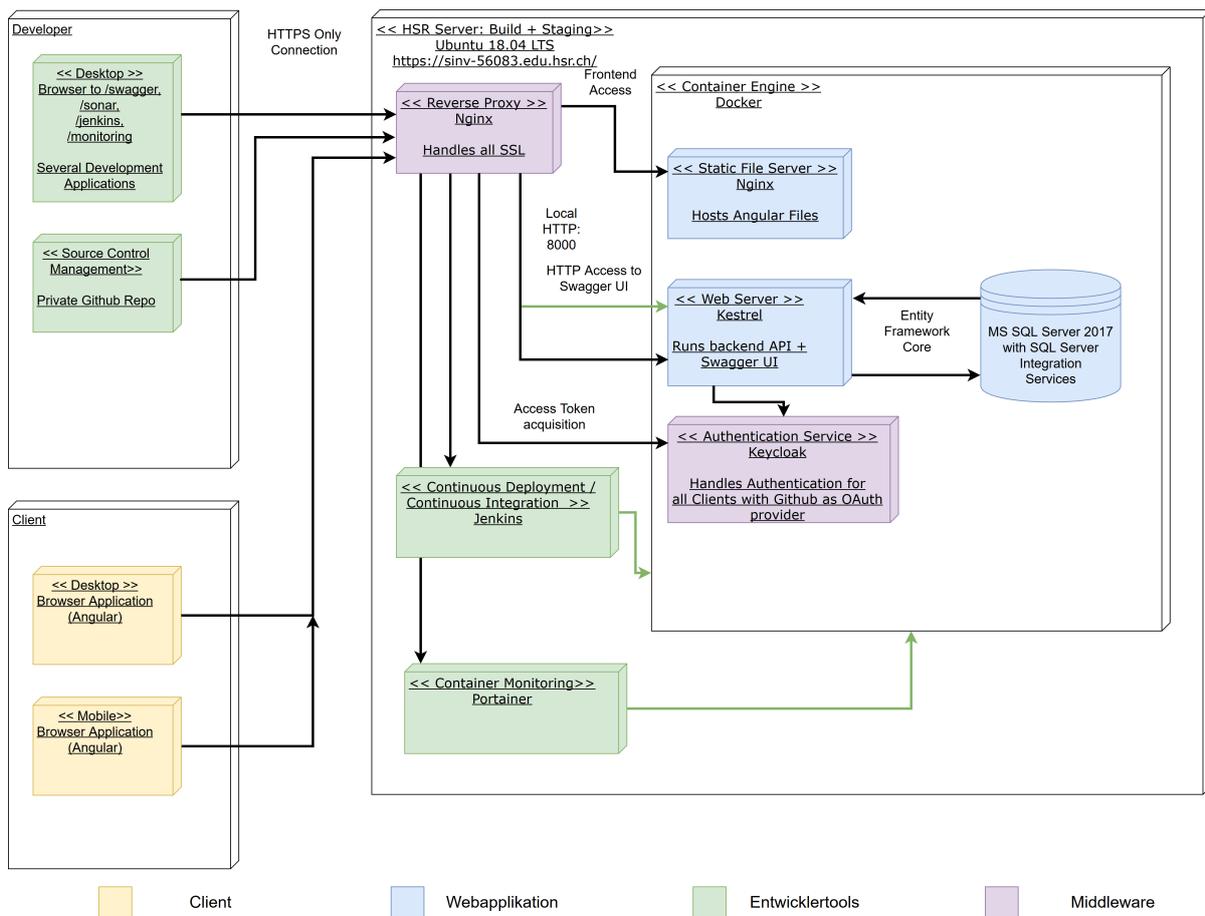


Abbildung 49. Deployment Diagram für das HSR Staging System

*Alle Kernkomponenten (Frontend, Backend, MS SQL, Keycloak) laufen in Docker Containern und kommunizieren über das virtuelle Docker Netzwerk miteinander. Alle Serveranfragen werden durch den Nginx Reverse Proxy geleitet, welcher auch für die Verschlüsselung des Datenverkehrs mit TLS zuständig ist. Dank Let's Encrypt ist der Server immer mit einem aktuellen HTTPS Zertifikat ausgestattet, und mit HTTPS-only wird eine stets verschlüsselte Kommunikation gewährleistet.*

*Die abgebildeten Docker Container werden bei jedem Build neu erstellt und gestartet. Für diesen Prozess ist Jenkins zuständig. Die deployten Docker Container können mit dem Web UI von Portainer überwacht werden. Die Middleware ist Software welche nicht von uns entwickelt wurde, aber zentral für den Betrieb sind. Nginx als Reverseproxy exponiert die lokalen Docker Ports an die korrekten URLs. Keycloak fungiert als Single Sign On Service und Identity Provider für Front- und Backend.*

*Die generierte Swagger Dokumentation wird zusammen mit dem Backend betrieben und ist unter einer eigenen Domain erreichbar. Dies erlaubt es Entwicklern, die API interaktiv zu erforschen. Als Clients kommen sowohl Desktop- wie auch Mobile Clients in Frage, wenn diese aktuelle Versionen verwenden.*

— [\[SA-8\]](#)

Diese Konfiguration aus der Studienarbeit wurde während der Bachelorarbeit um einen an unsere Bedürfnisse angepassten MS SQL Docker Container mit SSIS Unterstützung ergänzt. Das Staging System wird ohne Daten betrieben und dient der Überprüfung, ob lauffähige Deployment Artefakte erzeugt wurden. Sämtliche Daten der Produktivinstanz unterliegen den Datenschutzbestimmungen von Siemens und dürfen nur innerhalb des Siemens-Netzwerks verwendet werden.

## 4.6. Querschnittliche Konzepte

In diesem Kapitel möchten wir auf einige übergreifende Konzepte und Lösungsansätze eingehen, welche an mehreren Stellen relevant sind.

### 4.6.1. User Experience

Um ein einheitliches und benutzerfreundliches Bedienerlebnis für den Anwender zu erreichen, setzen wir auf das *Clarity Design System*. Die drei Grundpfeiler von Clarity sind die folgenden:

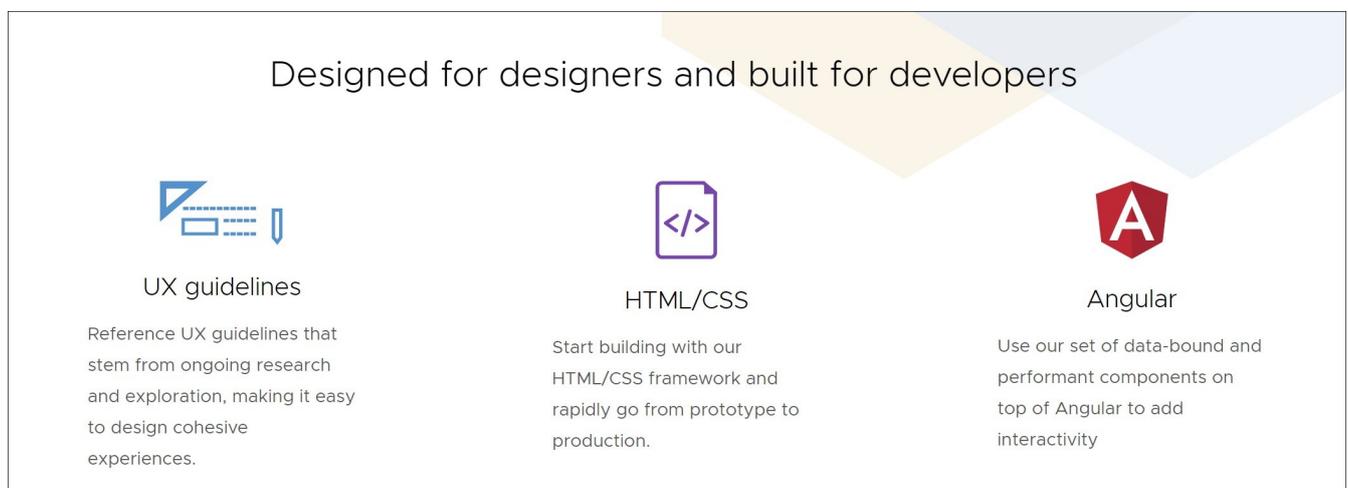


Abbildung 50. Die drei Grundpfeiler des Clarity Design System (Bildquelle: [\[Clarity-Intro\]](#)).

Der Einsatz von Angular als Frontend Technologie ermöglicht es uns, alle drei Grundpfeiler von Clarity einzusetzen. Somit können wir die zur Verfügung stehenden Funktionen optimal ausnutzen. Da bei der Ausarbeitung und Gestaltung von Benutzeroberflächen sehr viel Wissen erforderlich ist, insbesondere wenn es um den Zugang für Menschen mit einer Behinderung geht, sind bereits vorgegebene Gestaltungsrichtlinien sehr hilfreich bei der Umsetzung.

*Das Einhalten dieser Richtlinien ermöglicht es uns, eine Benutzeroberfläche zu gestalten, welche wie aus einem Guss wirkt und sich für den Endnutzer immer gleich anfühlt. So werden Benutzer durch die Applikation geführt, ohne von disruptiven UI Elementen verwirrt zu werden.*

— [\[SA-9\]](#)

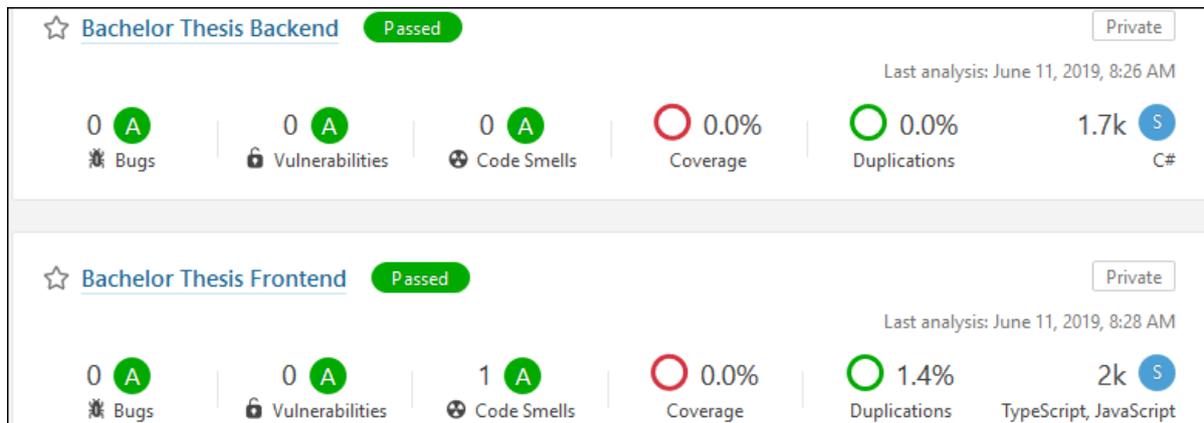
## 4.6.2. Quellcodegenerierung

Don't repeat yourself (DRY [\[PP\]](#)) ist ein Prinzip, welches besagt, Redundanz in Programmen möglichst zu vermeiden. Gerade bei verteilten Systemen, welche über APIs auf gemeinsame oder zumindest ähnliche Daten zugreifen, ist es schwierig, Redundanz komplett zu vermeiden. In dieser Arbeit wurde versucht, insbesondere *Representational Duplication* ([\[PP\]](#)) durch Generierung von Source Code zu minimieren. Dies hat es einerseits erlaubt, möglichst wenige Codeänderungen durchführen zu müssen und trotzdem API Anpassungen schnell durch das System zu propagieren. Andererseits hat dieses Vorgehen ermöglicht, alle APIs so einheitlich wie möglich zu halten. Hauptsächlich wurden hierfür drei Technologien verwendet:

1. **Entity Framework Core** bietet bei der Datenbankbindung mit dem Code First ([\[EF-1\]](#)) Vorgehen die Option, mit sogenannten *Migrations* ([\[EF-2\]](#)) sowohl Datenbanken zu erstellen, also auch diese nach Änderungen am Schema zu migrieren. Der dafür geschriebene Code kann direkt für das ORM Mapping zwischen der SQL Datenbank und den erstellten Modellklassen verwendet werden.
2. **NSwag** ([\[Nswag\]](#)) ist eine Swagger / OpenAPI Library ([\[OpenAPI\]](#)), welche es unter anderem ermöglicht, aus annotierten Controllerklassen Angular Services zu generieren. Somit ist es nicht nötig, Code für den API Zugriff zu schreiben, da dieser grösstenteils automatisch erzeugt wird. Gleichzeitig ist es möglich, die Dokumentation der öffentlichen API automatisch bereitzustellen.
3. **AutoMapper** ([\[AM\]](#)) ist eine Objekt-Mapping Library, welche das Umwandeln zwischen verschiedenen Objekten mit ähnlichen Daten erleichtert. So können Daten zwischen dem Model und Data Transfer Objects leicht ausgetauscht werden, ohne dass eine Umwandlung manuell erstellt werden muss.

### 4.6.3. Qualitätssicherung

Für regelmässige Code Reviews wurde Sonar als Inspektionswerkzeug gewählt. Die gewählte Konfiguration basiert auf der als "Strict" bezeichneten Vorkonfiguration, wobei sämtliche Regeln für die verfügbaren Sprachen angewandt werden. Wo sinnvoll wurden Regeln deaktiviert, da gewisse Regeln entweder zu stringente Anforderungen an Codekonstrukte haben, oder weil sich diese zum Teil widersprechen. Es wurden aktuell nur wenige Integrationstests für das Backend implementiert, um zu sehen, ob die Deployment Artefakte lauffähig sind. Leider funktioniert die Test Coverage von Sonar für diese jedoch nicht.



### 4.6.4. Build Historie

Da Jenkins für Continuous Integration verwendet wurde, lässt sich eine umfassende Übersicht über die Build Historie von Studien- und Bachelorarbeit anzeigen. Da während der Bachelorarbeit die Build Pipeline angepasst wurde um den neuen Deployments gerecht zu werden, sind nur die Build Vorgänge seit dem fertigen Deployment bei Siemens zu sehen. Jeder *Push* respektive *Pull* auf den Master Branch startet die Tests and Analysis Pipeline, welche wiederum bei Erfolg Docker Staging Deployment auslöst. Auch die Dokumentation wurde auf diese Art stets neu erstellt und stellt diese bereit.

S	W	Name ↓	Last Success	Last Failure	Last Duration
🌐	☀️	<a href="#">BA Docker Staging Deployment</a>	4 hr 7 min - <a href="#">#143</a>	12 days - <a href="#">#123</a>	3 min 51 sec
🌐	☀️	<a href="#">BA Test and Analysis</a>	4 hr 10 min - <a href="#">#90</a>	12 days - <a href="#">#68</a>	2 min 38 sec
🌐	☀️	<a href="#">BA-Dokumentation</a>	59 min - <a href="#">#260</a>	3 mo 9 days - <a href="#">#3</a>	3 min 12 sec
🌐	☀️	<a href="#">HSR-SA</a>	3 mo 8 days - <a href="#">#289</a>	5 mo 22 days - <a href="#">#275</a>	8 min 8 sec
🌐	☀️	<a href="#">SA-Dokumentation</a>	4 days 21 hr - <a href="#">#188</a>	5 mo 24 days - <a href="#">#101</a>	1 min 0 sec

## 4.6.5. Containerization

Die Entscheidung, Docker basierte Container als Deployment-Plattform für Entwicklung und Staging zu verwenden, hat sich aus mehreren Gründen vorteilhaft erwiesen. Nach der Studienarbeit wurden die Rahmenbedingungen für die Bachelorarbeit bestimmt, wobei ein zentraler Punkt die Integration in das Siemensnetzwerk dargestellt hat. Diese Entscheidung hat es unausweichlich gemacht, mit Siemens-Rechnern im Siemens-Netzwerk zu arbeiten, weswegen die Entwicklungsarbeit zu grossen Teilen nicht mehr mit privaten Maschinen erledigt werden konnte.

*Die Möglichkeit, reproduzierbare Deployments auf verschiedenen Maschinen zu garantieren ermöglicht es den Entwicklern, so nah wie nur möglich am Produktivsystem zu entwickeln. So werden viele Probleme bei der Abstimmung von Systemkomponenten eliminiert und erlauben den Fokus auf die eigentliche Entwicklung.*

— [SA-6]

Docker hat die reibungslose Migration auf andere Entwicklungsmaschinen ermöglicht. Identische Docker Compose Konfigurationen konnten verwendet werden, um in kurzer Zeit lauffähige Entwicklungsumgebungen bereitstellen zu können. So wurde das zeitaufwendige Einrichten zahlreicher Komponenten vermieden, womit mehr Zeit für die eigentliche Entwicklung aufgewendet werden kann. Gleichzeitig hat dies demonstriert, dass die Erweiterung des Teams technisch sehr einfach möglich wäre. Da sämtliche Systemkonfigurationen als Textdateien vorliegen können diese auch leicht in der Versionskontrolle abgelegt werden.

*Gleichzeitig wäre es mit dem aktuellen Stand [Anm: Ende der Studienarbeit, als das fertige Docker Deployment erarbeitet war] möglich, die Applikation mit allen Komponenten in kurzer Zeit auf ein anderes System zu migrieren und in Betrieb zu nehmen.*

— [SA-6]

Durch die Vorgabe das Deployment nativ (sprich ohne Docker) auf Windows Server vorzunehmen, wurde es notwendig die in den *Dockerfiles* vorgenommenen Konfigurationen manuell an den Komponenten vorzunehmen. Da alle Zusätze, womit fertige *Dockerfiles* erweitert werden, als explizite Konfigurationszeilen hinterlegt werden müssen, ist es aus den Dateien möglich, alle Konfigurationsschritte, welche für ein Deployment nötig werden, nachzuvollziehen. Ferner sind auch alle Abhängigkeiten der in Docker Containern

laufenden Applikationen auf diese Weise explizit über die *Dockerfiles* festgelegt, womit es kaum versteckte, fehlende Abhängigkeiten zu suchen gilt. Der folgende Auszug zeigt die benötigten Schritte, welche für das Produktiv-Deployment von Keycloak bei Siemens vorgenommen werden mussten.

*Beispielkonfiguration der Keycloak Stagingkonfiguration (Auszug)*

```
1 # Benoetigte Keycloak Version
2 FROM jboss/keycloak:4.8.3.Final
3 USER root
4 # Explizite Abhaengigkeit
5 RUN yum install -y zip xmlstarlet
6 # Benoetigte Dateien
7 COPY keycloak-theme /tmp/theme
8 # Entpacken des Keycloak Themes
9 RUN cd /tmp/theme && zip -r sep-theme *
10 # Benoetigter Keycloak User
11 USER jboss
12 RUN keycloak/bin/jboss-cli.sh
13 # Hinzufuegen von Theme Modul
14     --command="module add --name=sep --resources=/tmp/theme/ sep-theme.zip"
15 # Anpassung von XML File fuer Themebenutzung. Verkuerzt.
16     && xmlstarlet ed --inplace -P \
```

Bei der Inbetriebnahme der Komponenten hat sich die Aussage bewahrheitet, und es mussten mit Ausnahme der Backend-Applikation keine Konfigurationen speziell angepasst, sondern nur angewandt werden. Für Kestrel (*Standard HTTP Server für dotnet core Applikationen*) entwickelte Applikationen sind in der Regel ohne zusätzliche Konfiguration mit IIS aufzusetzen. Die Verwendung von Standardkomponenten hat es auch hier erlaubt, eine einfache Inbetriebnahme zu garantieren.

## 4.7. Risiken

Die folgenden Risiken wurden im bisherigen Verlauf der Entwicklung identifiziert. Der Stand der Risiken entspricht dem aktuellen Projektstand.

### 4.7.1. [R1] Temporal Tables

**Status:** Strategie zur Risikominimierung definiert.

#### Beschrieb

Bei VCR gibt es neben den aktuellen Daten auch historische Daten. Mit neuen Versionen von MS SQL gibt es Möglichkeiten zur einfachen Rückverfolgbarkeit von historischen Daten, mit den sogenannten *Temporal Tables* ([\[MS-SQL-Temporal-Tables\]](#)). Der Einsatz dieser würde viele Vorteile erbringen und ist unser bevorzugter Lösungsansatz. Falls das nicht möglich ist braucht es zwingend ein anderes Konzept und höchstwahrscheinlich Datenmodelländerungen.

#### Risiken & Folgen

Falls sich *Temporal Tables* wider Erwarten aus Gründen der technischen Limitation nicht einsetzen lassen, muss für die Modellierung von historischen Daten ein neuer Ansatz gefunden werden. Als Folge lassen sich keine Aussagen darüber machen, wie der Datenbestand zu einem bestimmten Zeitpunkt in der Vergangenheit aussah. So würden dann beispielsweise kein Audit-Trail und keine Trend Daten zur Verfügung stehen.

#### Strategie zur Risikominimierung

- Die technische Machbarkeit (Setup, Integration in Tooling, ...) soll frühzeitig durch einen Proof of Concept sichergestellt werden.
- Die ETL Jobs sollen dazu genutzt werden, die Daten regelmässig zu *überschreiben* und somit schnell eine Datenhistorie aufzubauen, um den realen Nutzen testen zu können.
- Der Meilensteinbewilligungsprozess soll von Anfang an auf diesen historischen Daten aufbauen, um allfällige Probleme frühzeitig zu erkennen.

## 4.7.2. [R2] Performanz im Parallelbetrieb

**Status:** Strategie zur Risikominimierung definiert und teilweise umgesetzt.

### Beschrieb

Da das neue Tool über längere Zeit parallel zu VCR laufen wird und VCR während dieser Zeit weiterhin als Datenmaster agiert, müssen die Daten zwischen den Tools ausgetauscht und konsistent gehalten werden. Der Anwender soll bei der Verwendung der Software keine Leistungseinbussen feststellen. Gleichzeitig darf die Datenkonsistenz keinesfalls auf Kosten von Leistungsoptimierungen gehen.

### Risiken & Folgen

Schlechte Performanz führt zu Frustration bei den Anwendern und wirkt sich negativ auf die Reputation und Akzeptanz der Software aus. Dateninkonsistenz führt rasch zu Vertrauensverlust in die Applikation. Dies hat oft zur Folge, dass Anwender beginnen, selbst die Daten auch noch zu tracken und damit eine Doppelbuchhaltung gestartet wird.

### Strategie zur Risikominimierung

- Die Performanz der ETL Jobs muss regelmässig gemessen und allenfalls optimiert werden.
- Die Datenkonsistenz muss kontinuierlich geprüft und bei allfälligen Problemen unverzüglich gehandelt werden.
- Beim Datenmapping zwischen dem alten und neuen System soll mindestens nach dem Vieraugenprinzip vorgegangen werden, um das Risiko von Fehlern im Datenmapping zu minimieren.

### Bereits umgesetzte Massnahmen

- Bisherige Performanzprobleme (z.B. beim Laden der Projektliste und des Hierarchiebaums) wurden mit entsprechenden Optimierungen in der Applikation gelöst.
- ETL Jobs, welche nach der ersten Implementierung sehr langsam waren, wurden in einem späteren Schritt drastisch optimiert.

### 4.7.3. [R3] Sicherheitsanforderungen mit Architektureinfluss

**Status:** Strategie zur Risikominimierung definiert.

#### Beschrieb

Der Trend zu immer höheren Sicherheitsanforderungen zeigt sich aktuell in vielen Applikationen. So hat die Verbreitung von *Multifaktor-Authentisierung* in letzter Zeit stark zugenommen. Auch die digitale Signierung (z.B. bei der Meilensteinbewilligung) erachten wir als ein realistisches Szenario. Im Moment sind noch keinerlei solcher Anforderungen bekannt. Sind bedeutende Verschärfungen am Sicherheitskonzept notwendig, so können diese unter Umständen auch Einfluss auf die Architektur haben.

#### Risiken & Folgen

Werden geänderte Sicherheitsanforderungen erst während der Entwicklung bekannt, können daraus folgende notwendige Änderungen an der Software zu Verzögerungen bei der Auslieferung führen.

#### Strategie zur Risikominimierung

- Die verschiedenen Möglichkeiten für Authentifizierung und deren Machbarkeit sollen unseren Partnern bei Siemens aufgezeigt werden, um ihnen eine Entscheidungsgrundlage zu geben.
- Die Vorgaben zu Services und Technologien von Siemens sind unbedingt zu beachten.

#### 4.7.4. [R4] Schlechte Akzeptanz der Benutzeroberfläche

**Status:** Strategie zur Risikominimierung definiert und teilweise umgesetzt.

##### Beschrieb

Die bisherigen Nutzer haben sich an die heutigen Abläufe in VCR gewöhnt. Neue Abläufe oder Darstellungen können eine Chance sein, aber auch ein Risiko, da die Macht der Gewohnheit bei solchen Werkzeugen nicht unterschätzt werden darf.

##### Risiken & Folgen

Zu hohe Komplexität, unklare Benutzerführung oder fehlende Übersicht führen dazu, dass Benutzer mit der Applikation nicht klar kommen. Je nach Ablauf sucht sicher der Nutzer entweder die Informationen wo anders falls vorhanden, die Verwendung führt zu einem negativen Gefühl oder Frust und kann zu Vertrauensverlust führen, falls der Nutzer das Gefühl hat, dass die Daten nicht stimmen.

##### Strategie zur Risikominimierung

- Diverse Probeläufe mit Stakeholdern und Endanwendern machen, sobald das Produkt einen dafür vernünftigen Reifegrad erreicht hat.
- Bestehende UX Guidelines nutzen und allenfalls UX Feedback einholen von extern.
- Klar aufzeigen, aus welcher Datenquelle die dargestellten Daten kommen und wie deren Wert zu Stande kommt (z.B. ein Tooltip bei einer Zahl, welches Hintergrundinformationen zur Berechnung gibt).

##### Bereits umgesetzte Massnahmen

- Erste Probeläufe mit Stakeholdern wurde durchgeführt und Feedback grösstenteils adressiert.
- Bei UX stützen wir uns primär auf die Vorgaben des *Clarity Design Systems* und nutzen unsere Erfahrung, um allfällige Probleme direkt während dem Entwicklungszyklus zu erkennen und zu beheben.
- Ein externer UX Review wurde uns von unserem Auftraggeber für Anfang Mai versprochen, der Termin hat jedoch bis heute nicht stattgefunden.

### 4.7.5. [R5] Benötigte Daten lassen sich nicht einfach abgreifen

**Status:** Risiko minimiert und entsprechende Massnahmen grösstenteils umgesetzt.

#### Beschrieb

Bei der genaueren Analyse des heutigen *Milestone Declaration Sheet* hat sich gezeigt, dass sich dort eine beträchtliche Anzahl an Daten aus verschiedenen Teilbereichen des VCR Systems dargestellt werden. Zudem war aus den zugrunde liegenden Daten nicht klar, ob diese direkt aus der Datenbank direkt kommen oder ob es berechnete Werte sind.

#### Risiken & Folgen

Wir hatten die Befürchtung, dass viele der benötigten Daten und Zahlen nicht einfach so vom VCR abgegriffen werden können und wir schlimmstenfalls die Geschäftslogik zuerst mühsam mittels *Reverse Engineering* analysieren und nachprogrammieren müssen. Die Folgen davon wären fatal: Ein sehr hoher Entwicklungsaufwand, Fehleranfälligkeit und Redundanz, insbesondere falls dieselbe Geschäftslogik nachprogrammiert werden müsste.

#### Strategie zur Risikominimierung

- Mit dem Auftraggeber und Entwickler klären, wie die dem *Milestone Declaration Sheet* zugrunde liegenden Daten zu Stande kommen.
- Falls Daten nicht *Low Level* (also direkt aus der Datenbank) extrahiert werden können eine höherliegende Datenschnittstelle einfordern.

#### Bereits umgesetzte Massnahmen

Unsere Befürchtung hat sich nach eingehender Analyse des *Milestone Declaration Sheet* bestätigt. Ein Grossteil der Daten sind berechnete Werte, die bereits so ans heutige *Milestone Declaration Sheet* übermittelt werden. Die Entwickler von VCR haben uns innerhalb nützlicher Frist eine XML Schnittstelle zur Verfügung gestellt, wo wir die berechneten Werte abgreifen können.

- Wo möglich werden die Daten *Low Level* gemappt.
- Für die restlichen Daten wurde uns eine Schnittstelle zur Verfügung gestellt, welche wir bereits erfolgreich nutzen.
- Die restlichen Daten, Berechnungen und Konfigurationen, welche nicht direkt über eine Schnittstelle zur Verfügung stehen oder bisher gar nie in VCR waren, werden nun direkt in unserer Lösung vorgenommen und gehalten.

## 4.7.6. [R6] Datenkonsistenz zwischen den Systemen

**Status:** Strategie zur Risikominimierung definiert.

### Beschrieb

Datenkonsistenz ist in jedem System ein essenzieller Bestandteil. Bei diesem Szenario geht es darum, sicherzustellen, dass die Datenkonsistenz zwischen dem neuen System und VCR immer gewährleistet ist.

### Risiken & Folgen

Gerade bei einem kritischen Prozess wie dem Meilensteinbewilligungsprozess ist die Datenkonsistenz unerlässlich, da auf den zugrunde liegenden Daten weitreichende Geschäftsentscheide getroffen werden (z.B. ob ein Projekt weiterverfolgt, oder eingestellt wird). Falsche Daten könnten im schlimmsten Fall zu falschen Entscheidungen führen. Werden die Fehler in den Daten nicht bemerkt, kann dies dazu führen, dass die auf falschen Daten basierenden Entscheide zu Schlussfolgerungen führen, welche negativen Einfluss auf den Geschäftserfolg haben (Einhaltung von Termin, Wirtschaftlichkeit, ...).

### Strategie zur Risikominimierung

- Vor der Veröffentlichung von neuen Versionen soll immer mit einem Systemtest verifiziert werden, ob die Daten korrekt sind.
- Berechnungen von Werten sollen mit Unit Tests abgedeckt werden.
- Wo möglich sollen programmatische Konsistenzchecks gemacht werden.

## 5. Ergebnisse & Ausblick

Mit der durch uns umgesetzten Lösung konnten mehrere Mängel der bisherigen Lösung adressiert werden, insbesondere in Bezug auf Benutzerfreundlichkeit und Performanz. Die an einer Meilensteingenehmigung beteiligten Personen können nun, je nach Rolle welche Sie innehaben, den Verlauf einer Genehmigung mit einem einzigen Tool überwachen und steuern. Dies ist eine Verbesserung gegenüber der vorangehenden, teilweise instabilen Lösung, wo VCR, Excel und Email gleichermassen involviert waren. Die neue Webanwendung kann auch auf mobilen Geräten eingesetzt werden und ermöglicht deshalb neue Anwendungsszenarien. Das neue Genehmigungswerkzeug für Meilensteine ermöglicht eine konsistentere Datenverarbeitung und eine strengere Kontrolle des Projektfortschritts, ohne auf bisher gewohnte Funktionalität verzichten zu müssen.

Neben der Meilensteingenehmigung wurden zudem auch zahlreiche andere Funktionen aus dem Prototypen überarbeitet und wo notwendig einige neue Funktionen hinzugefügt. Die umgesetzte Funktionalität wird in den nachfolgenden Abschnitten detailliert erläutert und bebildert aufgezeigt. Da wir nicht überall eine komplette Umsetzung erreichen konnten, wird im Abschnitt [Zielerreichung & Umsetzungsgrad](#) die Erreichung der einzelnen Punkte noch einmal gesondert erörtert.

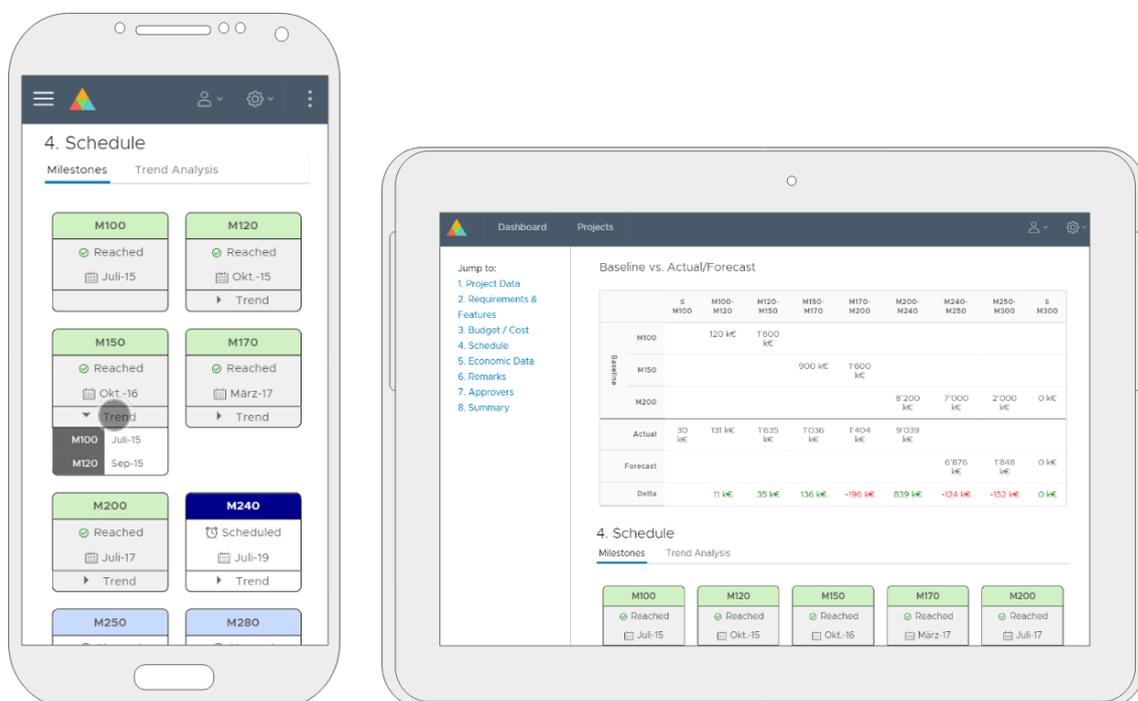


Abbildung 51. Erster Einblick in das Resultat der Bachelorarbeit: Alle für eine Meilensteinbewilligung notwendigen Projektkennzahlen werden gesammelt, aufbereitet und durch unsere neu entwickelte Applikation visualisiert.

## 5.1. Umgesetzte Funktionalität

### 5.1.1. Dashboard und Projektliste

Um zu einer konkreten Meilensteinbewilligung zu kommen, braucht es gewisse Basisfunktionen, wie zum Beispiel eine Übersicht über alle Projekte sowie eine Liste der Meilensteine für jedes Projekt. Diese Sichten wurden in einer ersten Version bereits in der vorangehenden Studienarbeit entwickelt. Geänderte Anforderungen, aber auch Überarbeitungen und Refactorings, beispielsweise um eine hohe Performanz der Projektliste zu erreichen, haben dafür gesorgt, dass wir an all diesen Benutzeroberflächen auch während der Bachelorarbeit gearbeitet haben.

#### Persönliches Dashboard

Sobald sich der Benutzer angemeldet hat, sieht er sein persönliches Dashboard. Dieses zeigt eine Liste von ausstehenden Meilensteinbewilligungen (sofern solche ausstehen), sowie eine Liste von Projekten, in welche er selbst involviert ist.

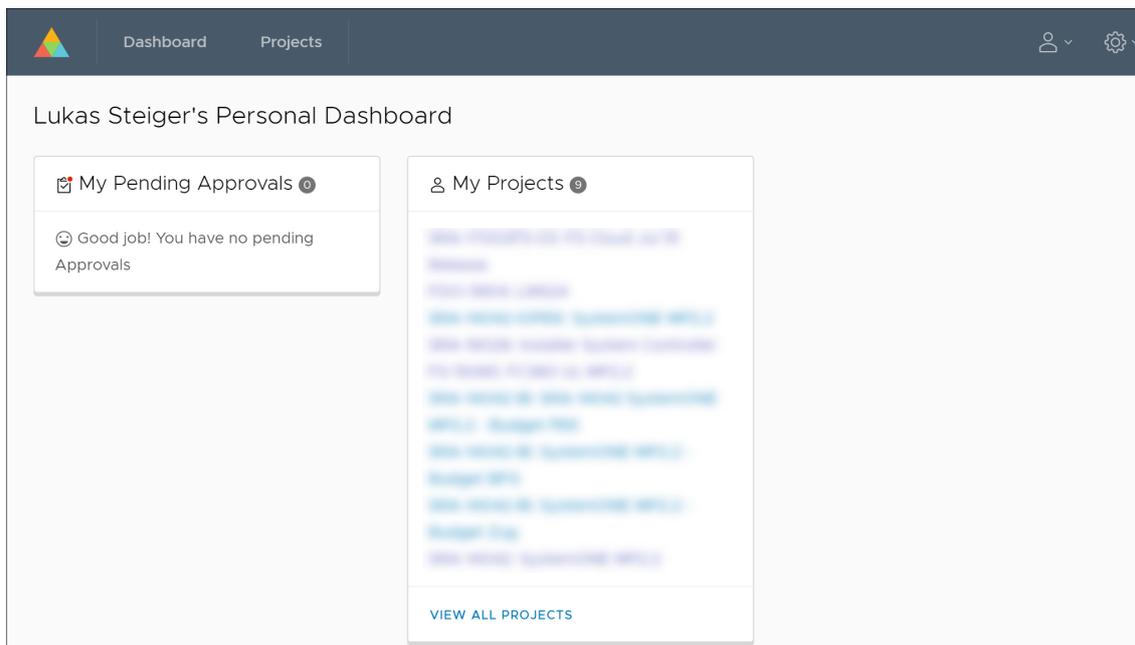


Abbildung 52. Persönliches Dashboard eines Benutzers

## Projektliste

Klickt man in der Hauptnavigation auf *Projects*, wird eine Projektliste sichtbar. Innerhalb dieser Liste werden tabellarisch sämtliche Projekte angezeigt, welche es in der auf der linken Seite selektierten Organisationseinheit gibt. Es kann beliebig sortiert oder gefiltert werden.

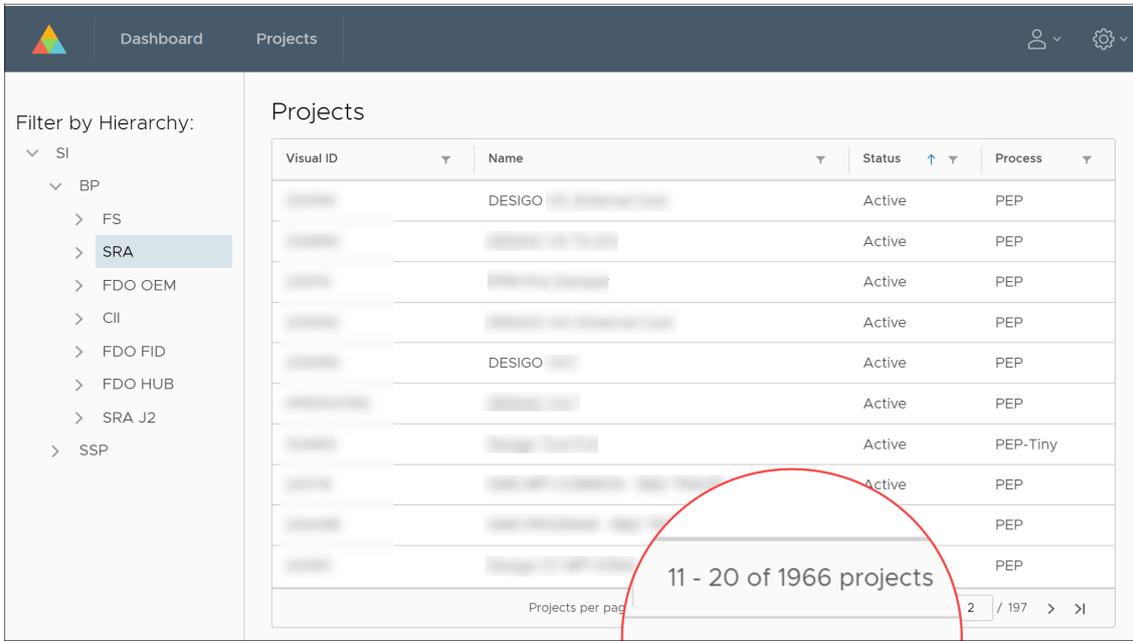


Abbildung 53. Alle 197 im Bild gezeigten Projektzeilen des ausgewählten Segments können angesehen, gefiltert und durchsucht werden.

Die zur Verfügung gestellten Sortier-, Filter- und Suchfunktionen sind blitzschnell, womit auch grössere Datenmengen problemlos clientseitig durchsucht werden können.

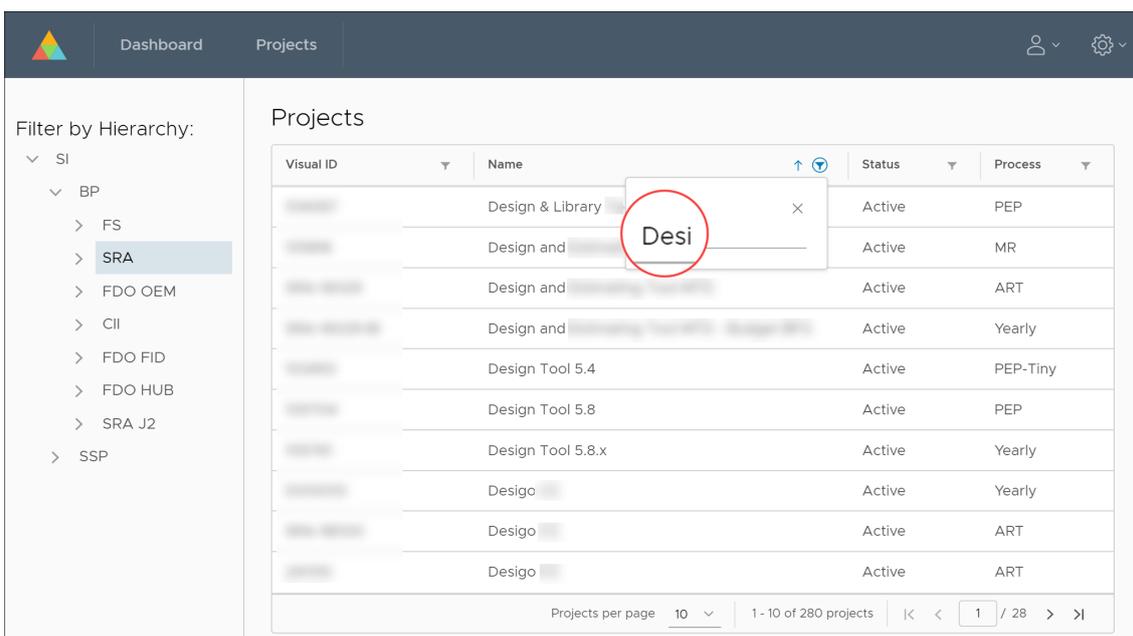
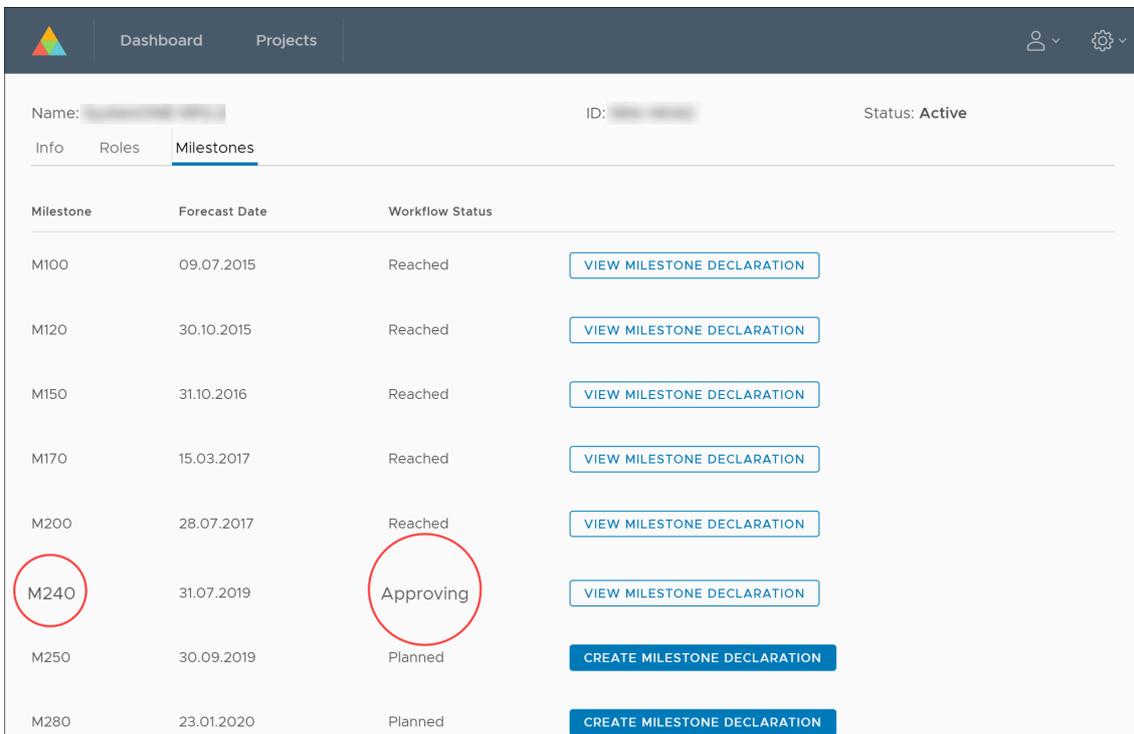


Abbildung 54. Volltextsuche innerhalb einer Projektliste

## Informationen über ein Projekt

Klickt man eine Projektzeile innerhalb der oben gezeigten Projektliste oder auf dem Dashboard in der *My Projects* Liste an, so landet man in der Projektübersicht. Diese Übersicht gibt es pro Projekt, wobei aktuell drei verschiedene Reiter zur Verfügung stehen. Im Reiter *Info* sind die wichtigsten Metadaten zum Projekt aufgelistet. Unter *Roles* sind verschiedene Projektrollen, bzw. welche Person die Rolle innerhalb dieses Projektes wahrnimmt, aufgelistet. Im dritten Reiter *Milestones* ist die Meilensteinliste sichtbar. Von hier kann ein Meilensteinbewilligungsprozess initiiert oder der Status eines sich im Bewilligungsprozess befindlichen Meilensteins nachverfolgt werden.



Milestone	Forecast Date	Workflow Status	
M100	09.07.2015	Reached	<a href="#">VIEW MILESTONE DECLARATION</a>
M120	30.10.2015	Reached	<a href="#">VIEW MILESTONE DECLARATION</a>
M150	31.10.2016	Reached	<a href="#">VIEW MILESTONE DECLARATION</a>
M170	15.03.2017	Reached	<a href="#">VIEW MILESTONE DECLARATION</a>
M200	28.07.2017	Reached	<a href="#">VIEW MILESTONE DECLARATION</a>
M240	31.07.2019	Approving	<a href="#">VIEW MILESTONE DECLARATION</a>
M250	30.09.2019	Planned	<a href="#">CREATE MILESTONE DECLARATION</a>
M280	23.01.2020	Planned	<a href="#">CREATE MILESTONE DECLARATION</a>

Abbildung 55. Meilensteinübersicht für ein Projekt

## 5.1.2. Meilensteindeklaration und Bewilligung

Unsere Hauptarbeit bestand in der Ablösung des Meilensteindeklarations- und Bewilligungsprozesses. Wir haben mit unserer Neuentwicklung im Wesentlichen das bisherige, auf Excel basierende *Milestone Declaration Sheet* abgelöst. Zu diesem Zweck haben wir die einzelnen Bestandteile der bisher genutzten Excel Datei auseinandergenommen und analysiert. Daraus resultierend haben wir zuerst einen Prototyp gemacht (siehe [Digitaler Prototyp](#)) sowie [Detaillierte Wireframes auf Papier](#). So konnten wir unsere Idee bereits früh festhalten und erstes Feedback einholen.

### Übersicht, Projektinformationen und Readiness Check

Die nachfolgend dargestellte Projektübersicht zeigt den Beginn der Meilensteindeklaration. Sie zeigt den zu bewilligenden Meilenstein, den verwendeten Geschäftsprozess sowie ein paar wichtige Metadaten. In der rechten Tabelle werden zudem standardmässig die wichtigsten fünf Rollen dargestellt. Es können aber auch sämtliche Rollen dargestellt werden.

Vorangängig oder parallel muss zudem noch der sogenannte *Milestone Readiness Check* durchgeführt werden. Das Ergebnis dieser Checkliste muss hier in der Auswahlliste "Milestone Readiness Check done?" entsprechend ausgewählt werden.

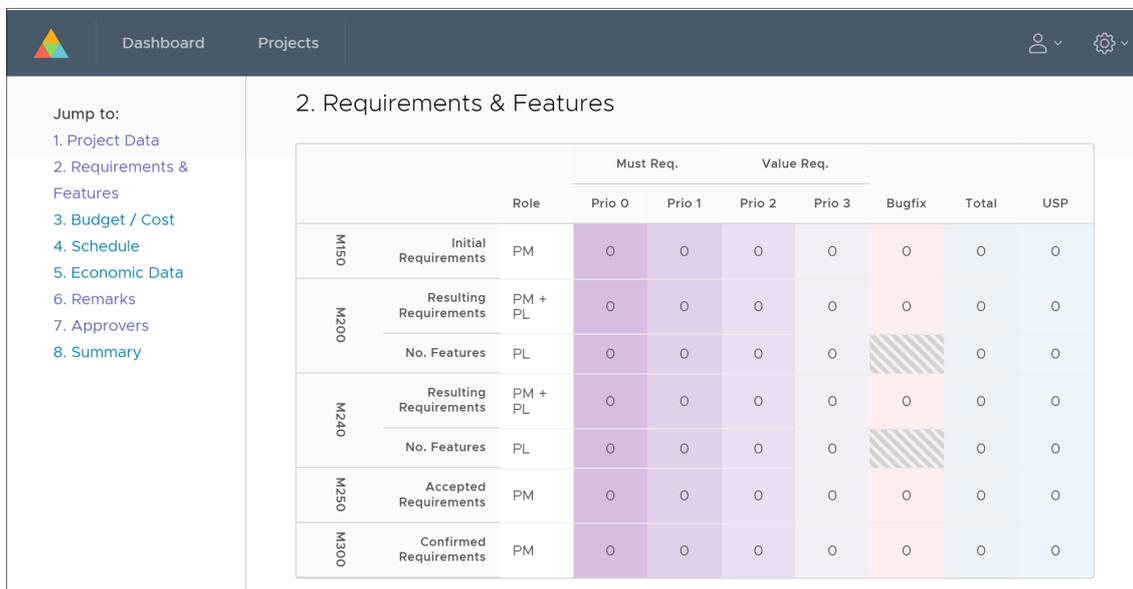
The screenshot displays the 'M240 Declaration' screen in the Siemens Smart Enterprise Project Cockpit. The interface is divided into a left sidebar with navigation options (1-8) and a main content area. The main content area features a 'M240 Declaration' header, a 'Milestone to declare based on process variant PEP' section with a row of milestone buttons (M100-M310), and a '1. Project Data' table. Below the table is a 'Milestone Readiness' section with a dropdown menu for 'Milestone Readiness Check done?' and a 'Checklist:' label.

1. Project Data	
Project Name	[Redacted]
Group ID / Location / Project ID	[Redacted] / ZUG / [Redacted]
Product Line	[Redacted]
Process Variant	PEP
Quality Gate	false
Milestone Date	31.07.2019
Project Leader	Steiger, Lukas
Product Manager	[Redacted]
Project Quality Manager	[Redacted]
PPL	[Redacted]
PPM	[Redacted]
<input type="checkbox"/> Show all Roles	

Abbildung 56. Darstellung der wichtigsten Metadaten des Projektes und Auflistung aller im Projekt involvierten Rollen.

## Anforderungen und Features

Als nächstes findet sich eine tabellarische Darstellung mit aggregierten Kennzahlen im Bereich der Anforderungsanalyse. Hier werden die Anzahl der Anforderungen (wie man sie aus klassischen Projekten kennt) sowie die Anzahl der Features (wie man sie aus agilen Projekten kennt) abgestuft nach Priorität dargestellt. Aus dieser Tabelle kann die Entwicklung der Anzahl Anforderungen entnommen werden, da die Kennzahlen bereits relativ früh im Projekt (M150) zum ersten Mal, gegen Ende des Projektes (M300) zum letzten Mal sowie mehrmals dazwischen erfasst werden. Oft führen eine steigende Anzahl Anforderungen zu einem verspäteten Liefertermin und zu höheren Kosten. Solche Entwicklungen lassen sich aus dieser sowie den nachfolgenden Darstellungen entnehmen.



		Role	Must Req.				Value Req.		Total	USP
			Prio 0	Prio 1	Prio 2	Prio 3	Bugfix			
M150	Initial Requirements	PM	0	0	0	0	0	0	0	
M200	Resulting Requirements	PM + PL	0	0	0	0	0	0	0	
	No. Features	PL	0	0	0	0		0	0	
M240	Resulting Requirements	PM + PL	0	0	0	0	0	0	0	
	No. Features	PL	0	0	0	0		0	0	
M250	Accepted Requirements	PM	0	0	0	0	0	0	0	
M300	Confirmed Requirements	PM	0	0	0	0	0	0	0	

Abbildung 57. Summe der Anforderungen & Features in verschiedenen Projektphasen. Da die Übertragung der entsprechenden Daten noch nicht funktioniert, erscheinen hier alle Zahlen mit dem Wert null.

## Budget und Kosten

Die nächste Sektion beinhaltet die wichtigsten Kennzahlen aus dem Bereich *Budget* und *Kosten*. Diese wenigen Kennzahlen erlauben es den im Meilensteinbewilligungsprozess involvierten Parteien, die finanzielle Gesundheit eines Projektes zu beurteilen. Die Ansicht *Baseline vs. Actual/Forecast* stellt dabei die ursprünglich geplanten und bewilligten Projektkosten (*Baseline*) den effektiv bisher im Projekt angefallenen Kosten (*Actual*) gegenüber. Da zukünftige Projektphasen noch keine angefallenen Kosten haben, werden die ursprünglich für diese Phase geplanten Kosten den nach aktueller Planung voraussichtlich anfallenden Kosten (*Forecast*) gegenübergestellt.

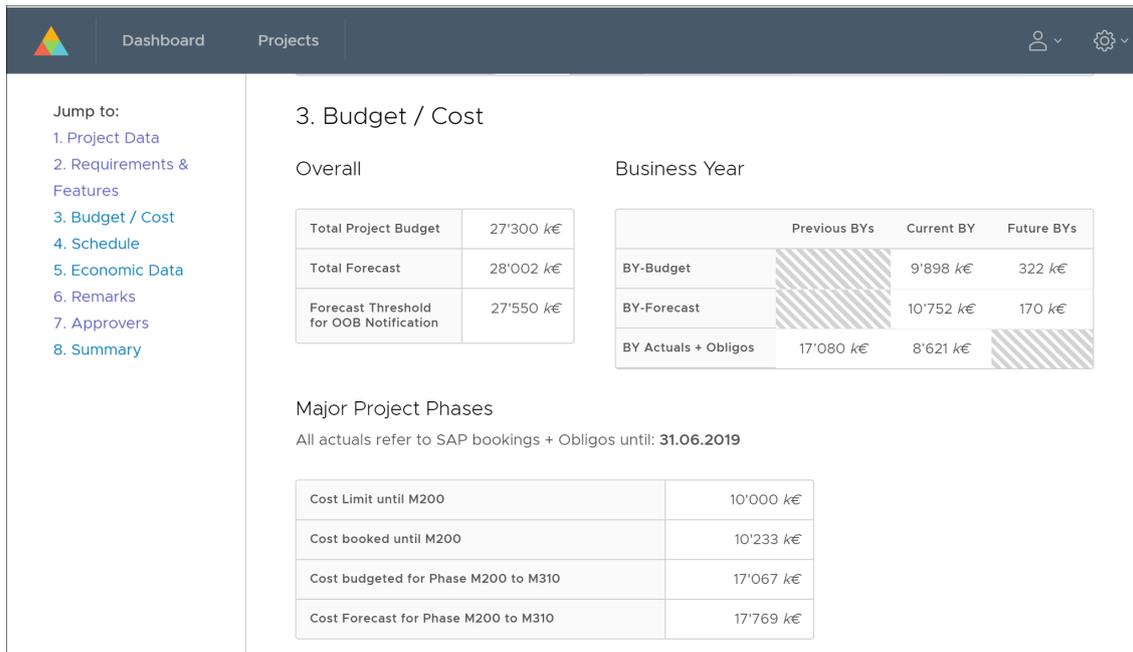


Abbildung 58. Übersicht über die wichtigsten Budget & Kosten Kennzahlen.

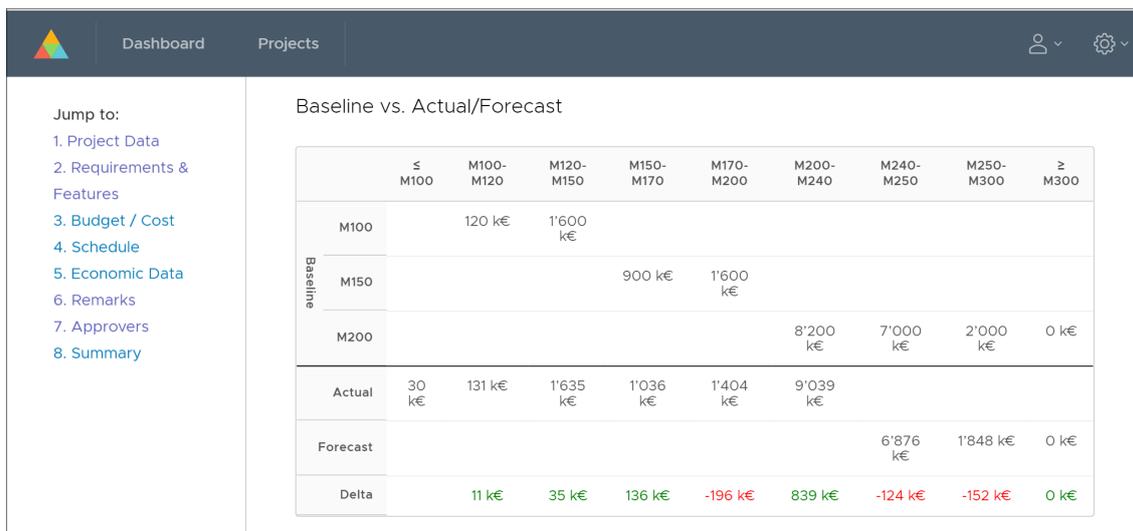


Abbildung 59. Vergleich des geplanten Budgets (Baseline) mit den effektiv angefallenen Kosten (Actual) bzw. den nach aktueller Planung voraussichtlich anfallenden Kosten (Forecast).

## Zeitplan

Als vierter Punkt folgt der Zeitplan. Hier werden im Reiter *Milestones* sämtliche Projektmeilensteine dargestellt. Bei erreichten Meilensteinen wird der effektive Termin der Meilensteinerreichung ausgewiesen, bei zukünftigen Meilensteinen das aktuell geplante Datum. Speziell hervorgehoben ist derjenige Meilenstein, für welchen wir gerade die Bewilligung ansehen.

Im zweiten Reiter *Trend Analysis* sollen die entsprechenden Daten graphisch als Meilensteintrendanalyse dargestellt werden.

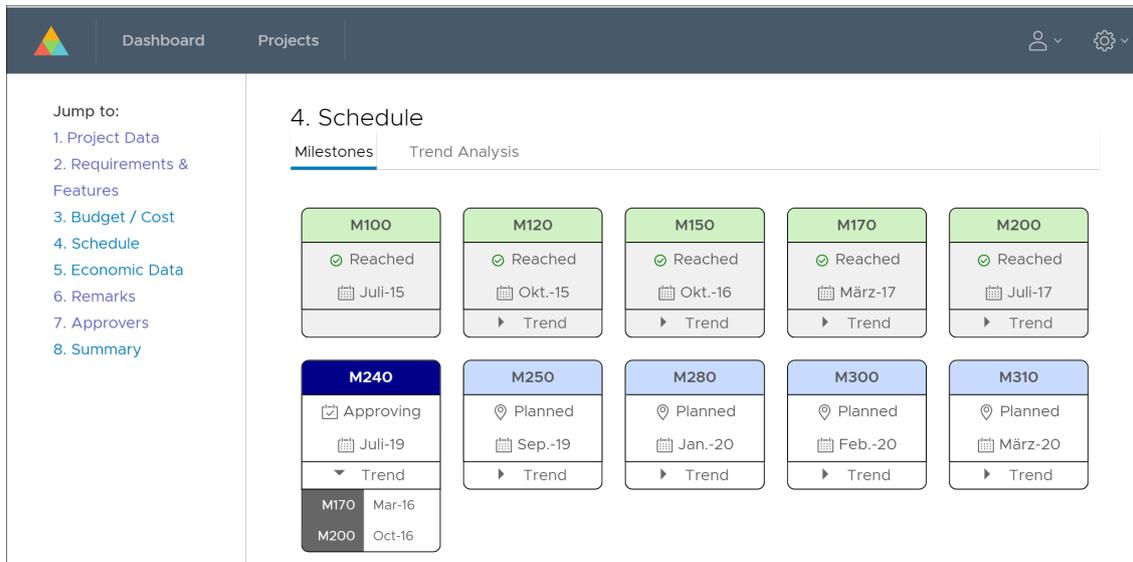


Abbildung 60. Die bisher erreichten und zukünftig anstehenden Meilensteine übersichtlich dargestellt. Durch einen Klick auf Trend können die ursprünglich geplanten Termine für den aktuellen Meilenstein (M240) angezeigt werden.

### Wirtschaftlichkeit und Bemerkungen

Danach folgt der fünfte Abschnitt, *Economic Data*, dessen Umsetzung wir aus Zeitgründen komplett weglassen mussten. Darauffolgend gibt es ein Kommentarfeld (*Remarks*), wo allgemeine Bemerkungen aber auch Erklärungen hinterlassen werden können.

### Bewilliger

Im zweitletzten Abschnitt sind sämtliche an dieser Meilensteinbewilligung beteiligten Personen aufgeführt. Wenn der Bewilligungsprozess bereits angestoßen ist, sieht man pro Person, ob die Rückmeldung bereits erfolgt ist und wer den Meilenstein bewilligt oder allenfalls auch abgelehnt hat.

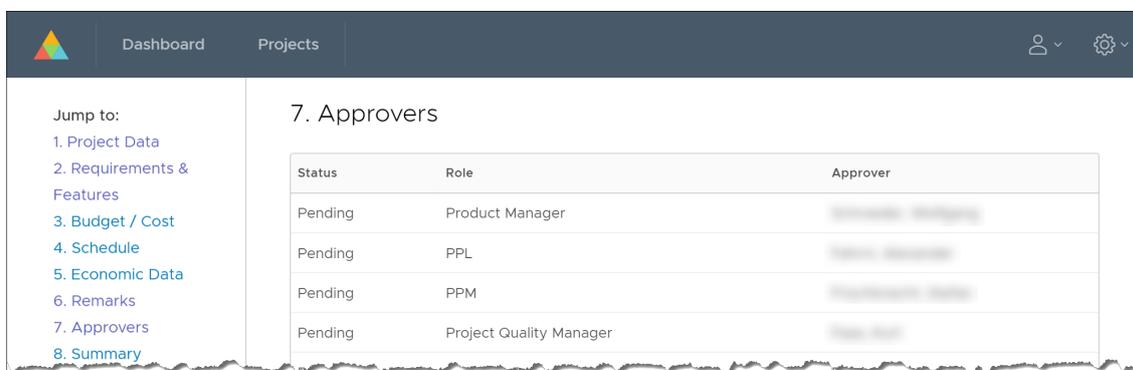


Abbildung 61. Auflistung sämtlicher Projektbeteiligten, welche für diesen Meilenstein zwingend ein Approval geben müssen.

## Zusammenfassung und Bewilligung

Zum Schluss soll nochmals ein Gesamtüberblick über den im Rahmen dieses Projektes zu bewilligenden Meilenstein gegeben werden. Die hier dargestellten Ampeln reflektieren jeweils den berechneten Status zu einem Teilaspekt aus der gesamten Meilensteindeklaration. So reflektiert beispielsweise die Ampel *Milestone Readiness Status* (in unserem Beispiel Gelb → d.h. teilweise erreicht) den vorangehend übertragenen Status aus der *Milestone Readiness Checklist* (weiter vorne in der Auswahlliste ist dieser auf "Yes - Criteria fulfilled with actions" gesetzt).

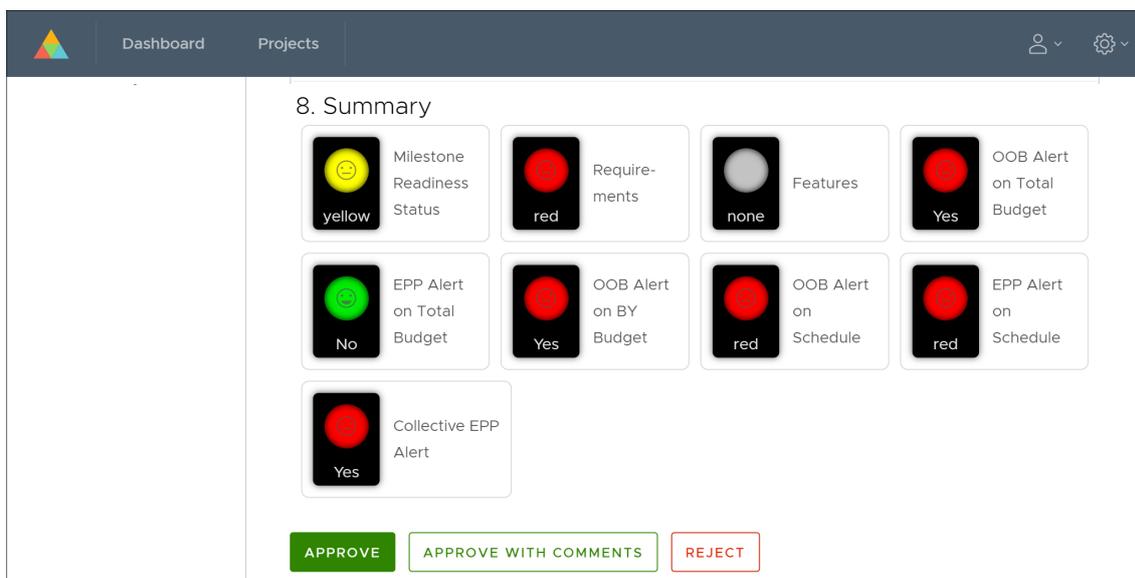


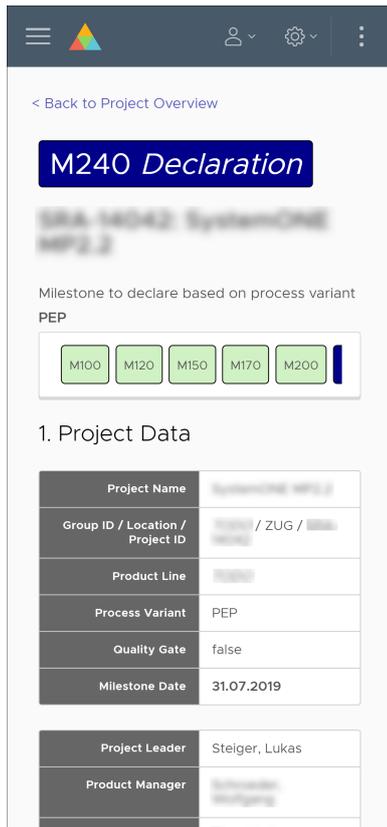
Abbildung 62. Die Ampeln reflektieren den Status der einzelnen Aspekte aus der Meilensteindeklaration. Im Optimalfall sind alle grün.

Durch einen Klick auf einen der Knöpfe *Approve*, *Approve with comment* oder *Reject* kann der Benutzer seine Rückmeldung zu dieser Meilensteinbewilligung ausdrücken.

### 5.1.3. Darstellung auf Mobilgeräten

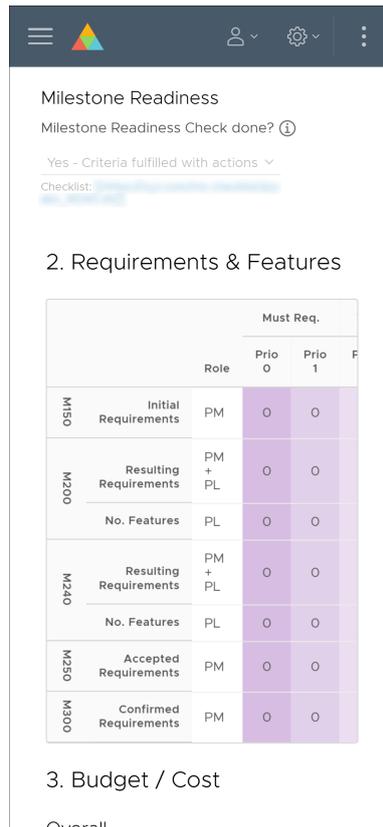
Um zu zeigen, dass der Meilensteinbewilligungsprozess auch auf mobilen Endgeräten funktioniert, sind die vorangehend bereits ausführlich erläuterten Schritte einer Meilensteinbewilligung noch einmal in der mobilen Ansicht aufgeführt.

Mobile: Projektübersicht



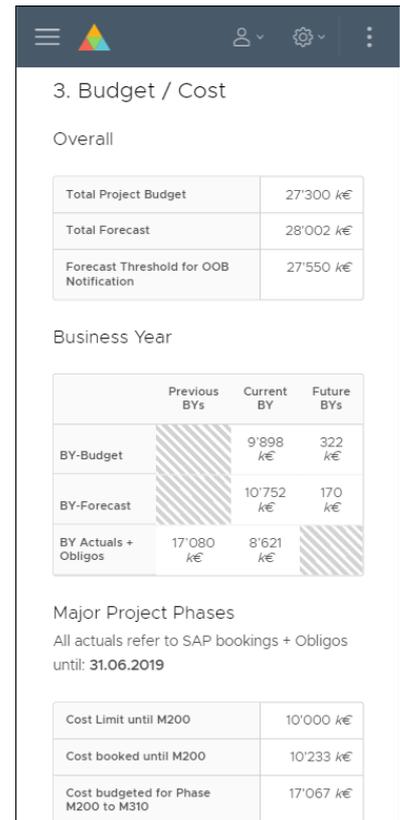
Einstieg und Übersicht über das Projekt, für welches eine Meilensteinbewilligung ansteht. Einige Metadaten und die wichtigsten Projektrollen werden dargestellt.

Mobile: Anforderungen



Tabellarische Darstellung der Anzahl und Priorität von Features & Anforderungen. Die Tabelle kann horizontal gescrollt werden. Dies erlaubt eine konsistente Darstellung auf verschiedenen Bildschirmgrößen.

Mobile: Budget

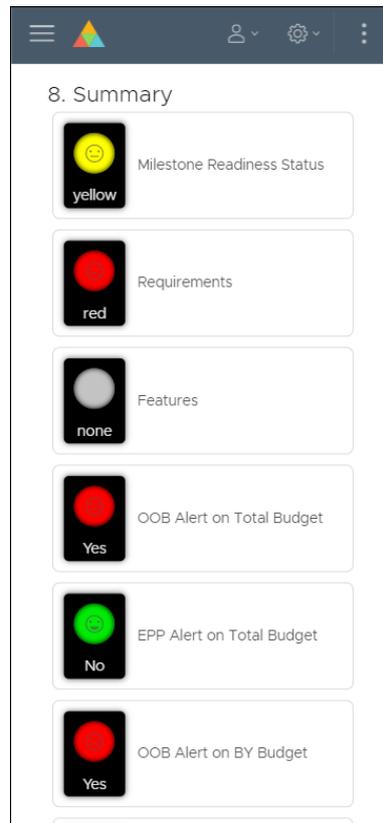


Die wichtigsten Budgetkennzahlen zum Projekt können ohne Einschränkung kompakt dargestellt werden. Die (hier nicht mehr dargestellte) Tabelle *Baseline vs. Actual/Forecast* ist ebenfalls mit horizontalem Scrolling umgesetzt.

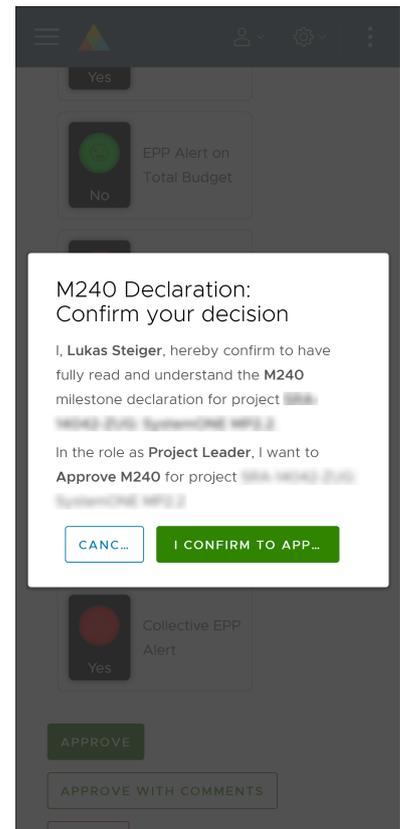
Mobile: Zeitplan



Mobile: Gesamtübersicht



Mobile: Bestätigungsdialog



Darstellung sämtlicher Meilensteine in diesem Projekt. Der Zeitpunkt der geplanten oder erfolgten Erreichung des Meilensteins sind ebenso dargestellt wie die zu einem früheren Zeitpunkt vorausgesagten Termine (Trend).

Status der einzelnen Teilaspekte. Die Darstellung des Status erfolgt als Verkehrsampel, um auf einen Blick den Erreichungsgrad erkennen zu können.

Bestätigungsdialog für einen Approver, wenn er einen Meilenstein bewilligen will, wie hier dargestellt. Die Ablehnung eines Meilensteins erfolgt nach dem gleichen Prinzip.

## Navigation auf mobilen Endgeräten

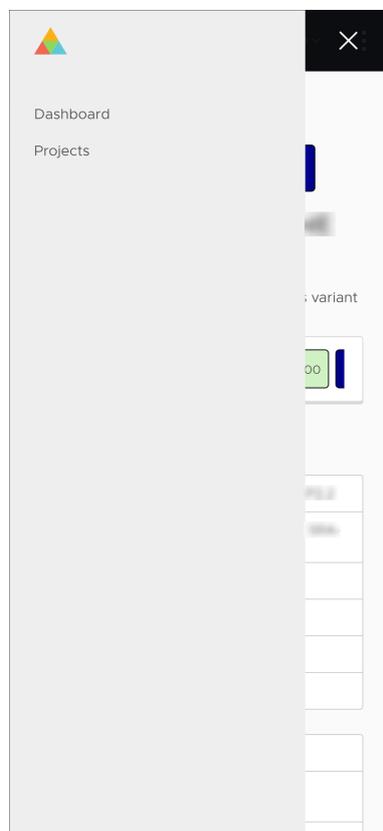
Die Darstellung der Haupt- und Unternavigation auf Endgeräten mit kleineren Bildschirmen ist nach den Gestaltungsvorgaben des *Clarity Design System* umgesetzt. Diese Richtlinien geben vor, dass Darstellungen auf Geräten, deren Breite eine bestimmte Mindestbreite unterschreiten, die Navigation ausgeblendet wird. Damit wird sichergestellt, dass auf kleineren Geräten der beschränkte Platz optimal ausgenutzt wird. Die ausgeblendeten Navigationselemente sind jedoch keinesfalls komplett verschwunden. Über einen Klick auf das entsprechende Menü-Symbol sind sie weiterhin erreichbar.

Mobile: Alle Navigations-  
elemente versteckt



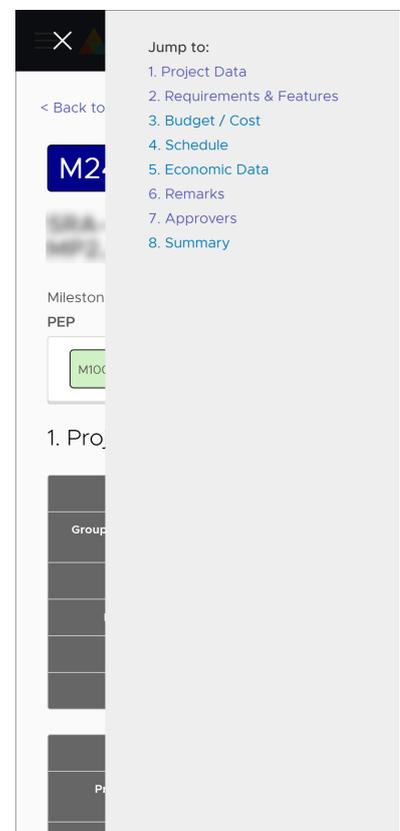
Als Vergleichsbeispiel hier nochmals eine bekannte Darstellung:  
Standardmässig sind die Navigationselemente in der mobilen Ansicht versteckt.

Mobile: Hauptnavigation  
ausgeklappt



Nach einem Klick auf das *Hamburger* Menu Icon oben links wird die Hauptnavigation eingeblendet.

Mobile: Unternavigation  
ausgeklappt



Ein Klick auf das Symbol mit den drei Punkten oben rechts öffnet die Unternavigation. Sie erlaubt es, innerhalb der Seite komfortabel hin und her zu springen.

## 5.2. Zielerreichung & Umsetzungsgrad

Die aktuell umgesetzten Funktionen sind in den beiden nachfolgenden Tabellen ausgewiesen.

Die erste Tabelle zeigt das Resultat unserer Hauptarbeit, den Meilensteinbewilligungsprozess zu digitalisieren. Die Ausweisung der umgesetzten Funktionalität erfolgt hier pro Abschnitt auf der Seite. Erläuterungen zum Umsetzungsgrad sind ebenfalls vorhanden.

*Tabelle 3. Umsetzungsgrad der einzelnen Teilbereiche der Meilensteinbewilligungsseite*

Abschnitt	Umsetzungsgrad	Erläuterungen
Übersicht	Komplett	<ul style="list-style-type: none"> <li>• Komplette Umsetzung während der BA.</li> </ul>
1. <i>Project Data</i>	Komplett	<ul style="list-style-type: none"> <li>• Komplette Umsetzung während der BA.</li> <li>• Die Auswahl aus der "Milestone Readiness Check done?" Auswahlliste wird automatisch gespeichert.</li> </ul>
2. <i>Requirements &amp; Features</i>	Nur Benutzeroberfläche	<ul style="list-style-type: none"> <li>• Benutzeroberfläche wurde während der BA entwickelt.</li> <li>• Datenanbindung ausstehend aufgrund der fehlenden <a href="#">Integration mit externen Schnittstellen</a></li> </ul>
3. <i>Budget / Cost</i>	Nur Benutzeroberfläche	<ul style="list-style-type: none"> <li>• Benutzeroberfläche wurde während der BA entwickelt.</li> <li>• Datenanbindung ausstehend aufgrund der fehlenden <a href="#">Integration mit externen Schnittstellen</a></li> </ul>
4. <i>Schedule - Milestones</i>	Komplett	<ul style="list-style-type: none"> <li>• Komplette Umsetzung während der BA.</li> <li>• Es fehlen noch die realen Trenddaten (siehe nachfolgende Zeile).</li> </ul>

Abschnitt	Umsetzungsgrad	Erläuterungen
4. <i>Schedule</i> - <i>Trend Analysis</i>	Keine Umsetzung erfolgt	<ul style="list-style-type: none"> <li>• Aus Zeitgründen wurde die Entwicklung der graphischen Darstellung einer Meilensteintrendanalyse zurückgestellt und konnte nicht mehr umgesetzt werden.</li> <li>• Datenanbindung ausstehend aufgrund der fehlenden <a href="#">Integration mit externen Schnittstellen</a></li> </ul>
5. <i>Economic Data</i>	Keine Umsetzung erfolgt	<ul style="list-style-type: none"> <li>• Aus Zeitgründen wurde die Entwicklung der Benutzeroberfläche zurückgestellt und konnte nicht mehr umgesetzt werden.</li> <li>• Datenanbindung ausstehend aufgrund der fehlenden <a href="#">Integration mit externen Schnittstellen</a></li> </ul>
6. <i>Remarks</i>	Komplett	<ul style="list-style-type: none"> <li>• Komplette Umsetzung während der BA.</li> <li>• Kommentar kann erfasst werden und wird automatisch gespeichert.</li> </ul>
7. <i>Approvers</i>	Komplett	<ul style="list-style-type: none"> <li>• Komplette Umsetzung während der BA.</li> <li>• Zeigt immer den aktuellen Bewilligungsstatus pro Benutzer an.</li> </ul>
8. <i>Summary</i>	Nur Benutzeroberfläche	<ul style="list-style-type: none"> <li>• Benutzeroberfläche wurde während der BA entwickelt.</li> <li>• Datenanbindung ausstehend aufgrund der fehlenden <a href="#">Integration mit externen Schnittstellen</a></li> </ul>

Abschnitt	Umsetzungsgrad	Erläuterungen
<i>Approve / Reject</i>	Komplett	<ul style="list-style-type: none"> <li>• Komplette Umsetzung während der BA.</li> <li>• Alle drei Funktionen können genutzt werden, ein Bestätigungsdialog stellt sicher, dass nicht ungewollt falsche Entscheidungen abgeschickt werden.</li> </ul>

Daneben haben wir an praktisch allen anderen Benutzeroberflächen, welche schon grösstenteils in einer ersten Version aus der Studienarbeit vorhanden waren, Änderungen vorgenommen. Der Umsetzungsgrad dieser ist nachfolgend ebenfalls ausgewiesen, wobei in dieser Tabelle die Gliederung pro Seite erfolgt.

*Tabelle 4. Umsetzungsgrad der einzelnen Teilbereiche im Rest der Applikation*

Seite	Umsetzungsgrad	Erläuterungen
Einstiegs-Dashboard (nicht eingeloggt)	Komplett	<ul style="list-style-type: none"> <li>• Wurde bereits während der SA umgesetzt.</li> <li>• Kleine Fehlerbehebungen während der BA.</li> </ul>
Delegiertes Login	Komplett	<ul style="list-style-type: none"> <li>• Wurde bereits während der SA umgesetzt.</li> <li>• Während der BA Neuinstallation bei Siemens.</li> </ul>
Persönliches Dashboard	Komplett	<ul style="list-style-type: none"> <li>• Erste Version wurde während der SA entwickelt.</li> <li>• Überarbeitung der Datenaufbereitung und Fehlerbehebungen während der BA.</li> </ul>
Projektliste	Komplett	<ul style="list-style-type: none"> <li>• Erste Version wurde während der SA entwickelt.</li> <li>• Komplette Neuimplementierung der Datenaufbereitung, für signifikant bessere Performanz (ungefähr 20x schnellere Ladezeiten).</li> <li>• Paging für die Tabelle im UI hinzugefügt.</li> <li>• Nicht mehr benötigte Funktionalität entfernt.</li> </ul>

Seite	Umsetzungs-grad	Erläuterungen
Projektansicht - Info	Komplett	<ul style="list-style-type: none"> <li>• Zeigt ein paar Metadaten zum Projekt an.</li> <li>• Bereits während der SA entwickelt und nur leicht angepasst.</li> </ul>
Projektansicht - Rollen	Komplett	<ul style="list-style-type: none"> <li>• Ansicht wurde während der SA entwickelt.</li> <li>• Komplettes Refactoring, entfernen der Editierfunktion, da nicht mehr gewollt.</li> </ul>
Projektansicht - Meilensteine	Fast komplett	<ul style="list-style-type: none"> <li>• Ansicht wurde während der SA entwickelt.</li> <li>• Während der BA grundlegend überarbeitet.</li> <li>• Einige Editierfunktionen aus der SA wurden entfernt, da nicht mehr gewollt.</li> <li>• Feature zur Aggregation der Anzahl Rückmeldungen pro Status und pro Meilenstein musste aus Performanzgründen temporär deaktiviert werden. Ansonsten voll funktionsfähig.</li> </ul>
Admin - Approval Matrix	Komplett	<ul style="list-style-type: none"> <li>• Visualisierung sämtlicher zur Verfügung stehenden Prozesse</li> <li>• Auflistung der Rollen, welche für diesen Prozess spezifiziert sind</li> <li>• Auflistung aller Meilensteine in diesem Prozess</li> <li>• Zuordnung (in Matrix-Form), welcher Rolle bei welchem Meilenstein in welchem Prozess ein <i>Approval</i> geben muss.</li> </ul>

## 5.3. Ausblick

Um die während der Arbeit entwickelte Applikation im Alltag einsetzen zu können, sind noch einige wenige Erweiterungen und Anpassungen nötig. Danach sollen durch Siemens weitere Arbeitsabläufe portiert werden, damit VCR mittelfristig Schritt für Schritt abgelöst werden kann. Die nachfolgenden Themengebiete sollen aufzeigen, wie das weitere Vorgehen aussehen könnte.

### 5.3.1. Meilensteinbewilligungsprozess ablösen

In Bezug auf die Applikation müssen noch die offenen Punkte im Meilensteinbewilligungsprozess, aufgeführt im Kapitel [Zielerreichung & Umsetzungsgrad](#), angegangen werden. Es wäre zudem vorteilhaft, wenn der Endstatus des Bewilligungsprozesses wieder direkt ins VCR zurückgeschrieben werden könnte. Mit der aktuellen Lösung muss dies noch manuell getan werden.

Im Bereich der Infrastruktur bei Siemens sind noch folgende Punkte umzusetzen:

- Die Applikation benötigt einen Domänennamen und ein gültiges Zertifikat.
- Die Konfiguration des Identity Providers muss angepasst werden.
- Die bisherige CI Infrastruktur muss zu Siemens portiert werden.

Danach steht der Ablösung des Meilensteinbewilligungsprozesses nichts mehr im Wege.

### 5.3.2. Change Requests implementieren

Es gibt einen zweiten Workflow, welcher ähnlich wie der Meilensteinbewilligungsprozess abläuft, der sogenannte OOB (Out of Bounds) Prozess. Sobald im Projekt eine Änderung an Budget oder Zeitplanung notwendig ist, weil diese ausserhalb eines gewissen Toleranzrahmens liegen, muss ein Änderungsantrag verfasst werden. Mehr dazu Im Kapitel [Budget & Schedule Change Request \(OOB\) Sheet](#). Aufgrund der ähnlichen Herausforderungen bietet es sich an, diesen Prozess ebenfalls rasch abzulösen, um eine weitere Schnittstelle eliminieren zu können.

### 5.3.3. Umgang mit historischen Daten

*Dank einer Schnappschussfunktion ist es heute in VCR bei einem beträchtlichen Teil der Projektdaten möglich, historische Daten in monatlichen Intervallen anzuschauen. Es fehlt jedoch ein Audit Trail, um Änderungen an Daten lückenlos nachvollziehen zu können.*

*Dank der Verwendung von MS SQL liesse sich die dort eingebaute Temporal Tables Funktion nutzen, mit Hilfe welcher sich gemäss unseren Recherchen Anforderungen wie die Nutzung eines Audit Trail oder die Möglichkeit zum zurückblättern zu einem Datenbestand an einem früheren Datum umsetzen liessen.*

— [\[SA-12\]](#)

### 5.3.4. Aggregierte Sichten

*Bei VCR [muss die Erzeugung eines] Reports manuell angestossen werden, um aggregierte Sichten auf mehrere Projekte zu erhalten. Die neue Applikation soll solche Sichten direkt in der Weboberfläche anbieten und der Benutzer soll die Möglichkeit erhalten, diese [individuell] an seine Bedürfnisse auszurichten (z.B. durch die Auswahl der einbezogenen Projekte). Dieser Wunsch wurde mehrfach in den [Interviews](#) geäussert.*

— [\[SA-12\]](#)

### 5.3.5. Anbindung an ein Reporting Tool

*VCR hat heute ein integriertes Reporting System, welches auf einer alten Version von Crystal Reports basiert. [...] In unserer neuen Lösung gibt es bis jetzt kein Reporting. Eine Möglichkeit wäre hier, eine kommerzielle Reporting / Business-Intelligence Lösung zu evaluieren und die neue Applikation daran anzubinden.*

— [\[SA-12\]](#)

### 5.3.6. Portierung der bisherigen Kernfunktion

Zuletzt steht noch der grösste Brocken an Arbeit an: Die bestehenden Kernfunktionen von VCR müssen in die neue Lösung portiert werden. Dies soll jedoch nicht etwa in einem Schritt geschehen, sondern in dem einzelne Sichten und Abläufe aus der heutigen Kernlösung herausgelöst und in der neuen Applikation neu umgesetzt werden. Wenn die Qualität und Isolation des bestehenden VCR Quellcodes stimmt, kann vorhandene Geschäftslogik einfach übernommen werden, da in beiden Systemen C# als Programmiersprache eingesetzt wird.

## 5.4. Schlussfolgerungen

Im Rahmen dieser Arbeit konnte gezeigt werden, dass es möglich ist, einen bestimmten Workflow aus einem Altsystem auszuwählen, zu isolieren und dafür einen Nachfolger zu entwickeln. Der hier verwendete Problemlösungsansatz ist ein Beispiel dafür, dass die schrittweise Modernisierung einer Anwendung ein praktikabler Ansatz ist, wenn man dabei sicherstellt, dass das alte System und die neue Anwendung während der Übergangszeit parallel betrieben werden können. Die Sicherstellung der Datenkonsistenz durch Implementierung von ETL Jobs hat sich als verlässliche Lösung herausgestellt. Diese Aspekte, verbunden mit einer benutzerfreundlichen, performanten neuen Lösung sorgen dafür, dass wir für Siemens eine gute Basis geschaffen haben, um VCR mittelfristig ablösen zu können.

Nun liegt es an unserem Auftraggeber Siemens, die letzten noch aus der Arbeit verbliebenen offenen Punkte umzusetzen (siehe [Zielerreichung & Umsetzungsgrad](#)). Da die Infrastruktur für die Applikation bereits betriebsbereit ist, kann die Applikation mit wenig Aufwand für Endanwender freigeschaltet werden.

Die Ablösung weiterer Abläufe unter Verwendung desselben Lösungsansatzes kann danach mit dem gleichen Ansatz vorangetrieben werden. So wäre es möglich, VCR innert weniger Jahre durch das neue System abzulösen, indem durch Anwendung einer agilen Vorgehensweise Schritt für Schritt Abläufe und Funktionen von VCR portiert werden. Dem Kapitel [Ausblick](#) sind dabei die wichtigsten Punkte zu entnehmen, welche in Zukunft umgesetzt werden sollen.

*Ein [Persönliches Fazit](#) der Autoren zum Ablauf und zur Umsetzung der Arbeit ist zudem dem Anhang zu entnehmen.*

## 6. Glossar



Einige der obigen Einträge im Glossar wurden eins zu eins oder in leicht abgeänderter Form aus der Studienarbeit übernommen. Zur besseren Lesbarkeit des Glossars haben wir auf einen Quellenverweis bei den einzelnen Einträgen verzichtet.

### Access Request

Antrag eines Benutzers, um Zugriff auf eine Applikation zu erhalten.

### Angular

Von Google entwickeltes und auf Typescript basierendes Frontend Webapplikationsframework. Siehe [\[Angular\]](#).

### Approval

Bezeichnet die Genehmigung/Zustimmung zu einer Sache. Im Kontext dieser Arbeit geht es, sofern nicht anders spezifiziert, um die formale Zustimmung zur Erreichung eines Projektmeilensteins.

### Approver

Person, welche aufgrund ihrer Rolle innerhalb einer Projektorganisation eine formale Zustimmung geben muss.

### ART

Agile Release Train. Ist ein langlebiges Team aus agilen Teams, welches, zusammen mit anderen Stakeholdern, inkrementell an einem Produkt arbeitet ([\[SAFE-ART\]](#)).

### BA

Bachelorarbeit - Bezeichnet diese Arbeit.

### CA

Certificate Authority. Herausgeber für digitale Zertifikate.

### Clarity Design System

UX Empfehlungen, HTML/CSS Framework und *Angular* Komponenten aus einer Hand, unter Führung von VMware. Mehr Infos siehe [\[Clarity\]](#).

## Corporate Entitlement Service (CES)

Zentraler *Identity Provider* bei Siemens. Siehe [\[Corporate Entitlement Service \(CES\)\]](#) für mehr Informationen.

## Daten Master

Bezeichnet diejenige Software Applikation, welche für die Korrektheit eines bestimmten Daten Set verantwortlich ist.

## DI

Steht für *Dependency Injection*. Entwurfsmuster zur zentral geregelten Abhängigkeitsauflösung von Objekten.

## ETL (Prozess)

*Extract, Transform, Load* (ETL) ist ein Prozess, bei dem Daten aus mehreren gegebenenfalls unterschiedlich strukturierten Datenquellen in einer Zieldatenbank vereinigt werden. *Extraktion* der relevanten Daten aus verschiedenen Quellen. *Transformation* der Daten in das Schema und Format der Zieldatenbank. *Laden* der Daten in die Zieldatenbank. [\[Wiki-ETL\]](#)

## ETL Tools

Programme, welche zur Ausführung von *ETL* Prozessen genutzt werden.

## HSR

*Hochschule für Technik Rapperswil*

## Identity Management (IM)

Identitätsmanagement ist der organisatorische Prozess zur Identifizierung, Authentifizierung und Autorisierung von Einzelpersonen oder Personengruppen für den Zugriff auf Anwendungen, Systeme oder Netzwerke, indem Benutzerrechte und -beschränkungen mit etablierten Identitäten verknüpft werden. (Deutsche Übersetzung der Definition aus [\[IM\]](#)).

## Identity Provider (IdP)

Applikation zur Verwaltung von Benutzeridentitäten. Stellt Services zur Benutzerverwaltung und Authentisierung zur Verfügung. Andere Applikationen können und sollen die Authentisierung an den *IdP* delegieren.

## Jazz (Plattform)

Teamkollaborationsplattform von IBM. Beinhaltet mehrere Applikationen für Requirements Engineering (RDNG), Arbeitsplanung und Fortschrittsverfolgung (RTC), Qualitätsmanagement und Testing (RQM) und Reporting (JRS).

## JRS

*Jazz Reporting Services*. Teil der *Jazz Plattform*. Webapplikation zur Konfiguration und Ausführung von Reports, basierend auf Echtzeit oder Data-Warehouse Daten.

## Meilenstein

Bezeichnet ein Etappenziel innerhalb eines Projektverlaufs. Das geht vom Start des Projekts über den Start der Entwicklung bis zur Marktreife. Anzahl, Inhalt und Umfang hängt stark vom gewählten *Projektprozess* ab.

## Meilensteinbewilligung / Meilensteinbewilligungsprozess

Bezeichnet den Ablauf, um die Erreichung eines Meilensteins formell vorzubereiten und die entsprechenden *Approvals* aller relevanten *Approver* einzuholen.

## Meilensteintrendanalyse

Die Meilensteintrendanalyse versucht, aus der zeitlichen Verschiebung von Meilensteinen eine Prognose für den weiteren Projektfortschritt und insbesondere für den Projektendtermin abzuleiten. Aus [\[Wiki-MTA\]](#).

## Milestone Declaration Sheet

Eine aus dem aktuellen Datenstand von *VCR* generierte Excel Datei, welche den Datenbestand zum Zeitpunkt einer anstehenden *Meilensteinbewilligung* widerspiegelt. Die Datei enthält einiges an Geschäftslogik und kann leider auch manipuliert werden.

## MS Approval

Steht für *Meilensteinbewilligung*.

## MS SQL (Server)

*Microsoft SQL Server*. Datenbankserver von Microsoft.

## Multifaktor-Authentisierung

Bezeichnet den Identitätsnachweis eines Benutzers durch die Kombination mehrerer unabhängiger Faktoren, z.B. ein Passwort, eine Smartcard oder ein *One Time Password* (SMS). Oft unter der Spezialisierung *Zwei-Faktor-Authentisierung* bekannt, wobei bei einer solcher immer exakt zwei Faktoren gefordert sind.

## MVC

Bezeichnet das allgemein bekannte *Model View Controller* Pattern. Im Kontext von *C#*, *.NET* oder *ASP* wird es oft als Abkürzung für das *ASP.NET MVC* Web Framework genutzt.

## OIDC

Steht für *Open ID Connect* und ist eine Authentifizierungsschicht, welche *OAuth 2.0* zur Grundlage hat. Damit kann die Identität eines Benutzers geprüft und grundlegende Information über den Benutzer abgerufen werden.

## On-Premise (Hosting)

Während bei einer *Cloud* basierten Software der Anbieter der Software oder ein Drittanbieter (z.B. AWS) den Betrieb verantwortet, wird bei *On-Premise* der Betrieb der Software durch den Kunden verantwortet.

## PL

Projektleiter

## PQM

Projektqualitätsmanager

## Projekt

Ein Projekt im klassischen Sinn. Hat oft Meilensteine. Im VCR wird alles als Projekt bezeichnet, obwohl es sich eigentlich um Projektzeilen handelt.

## Projektprozess

Beschreibt die Vorgehensweise bei der Umsetzung eines Projektes. Spezifiziert Projektrollen, geforderte Dokumente und Artefakte, Meilensteine und Quality-Gates.

## Projektzeile

Ein Teil eines Projektes. Ein Projekt, welches beispielsweise an drei Standorten entwickelt wird, hat (mind.) drei Projektzeilen, nämlich pro Standort eine Zeile.

**PSD (Meeting)**

*Project Status Decision (Meeting)*. Monatliches Meeting aller Entscheidungsträger, um pro Projekt wichtige Entscheidungen wie Meilenstein Approvals zu fällen, \_OOB\_s zu behandeln, den Erfolg des Projektes zu messen.

**RTC**

*Rational Team Concert*. Teil der *Jazz Plattform*. Webapplikation für Arbeitspaketverwaltung, Planung und Nachverfolgung.

**SA**

Studienarbeit - Vorarbeit "*NextGen Project Management Tooling*"

**SAML**

Steht für *Security Assertion Markup Language*. Wird in unserem Kontext im Zusammenhang mit *Identity Providers* genutzt, welche *SAML* als Protokoll zum Informationsaustausch nutzen. Ein alternatives Protokoll ist hier *OIDC*.

**Security Measurement Plan**

Bezeichnet einen Aktionsplan, der besagt, in welchem Rahmen und unter welchen Umständen ein Softwareprodukt in einem Unternehmen eingesetzt werden darf und welche Dinge dabei zu beachten sind. Betrachtet insbesondere aber keineswegs ausschliesslich Aspekte der Informationssicherheit.

**Siemens ID**

Alternativer *Identity Provider* bei Siemens. Siehe [\[Siemens ID\]](#) für mehr Informationen.

**SOAP**

Programmiersprachen und -modell unabhängiges Protokoll zum Austausch von strukturierten Daten, häufig im XML Format. Spezifikation und Quelle [\[SOAP\]](#).

**Social Login**

Darunter versteht man die Wiederverwendung von existierenden Benutzeraccounts für Soziale Netzwerke wie Facebook oder Google.

**SSIS**

Steht für *SQL Server Integration Services* und ist ein ETL Tool von Microsoft. Es ist Bestandteil vom Lieferumfang von *MS SQL Server*.

## Temporal Tables

*MS SQL* bietet ab der Version 2016 einige sehr nützliche Funktionen zur Analyse von historischen Daten, genannt Temporal Tables, siehe [\[MS-SQL-Temporal-Tables\]](#).

## Toollandschaft

Verschiedene Softwaretools, welche miteinander integriert sind, zusammen interagieren oder zumindest von den Anwendern zur Erfüllung ihrer Arbeitsabläufe zusammen genutzt werden.

## UX

User Experience, umfasst sämtliche Aspekte der Nutzererfahrung bei der Interaktion mit einem Softwareprodukt.

## VCR

Name der heute bei [\[Siemens\]](#) im Einsatz stehenden Projektverfolgungs-, Planungs- und Reporting Lösung. Soll Schritt für Schritt durch die von den Studenten im Rahmen der Bachelorarbeit entwickelte Lösung abgelöst werden. Die Abkürzung VCR steht für *Visualization, Controlling* und *Reporting*.

## 7. Quellenverzeichnis



Hinweis zur Verwendung von Zitaten im Text: Um einen optimalen Lesefluss in dieser Arbeit zu gewährleisten, wurden Schreibfehler aus Originalzitaten korrigiert sowie Anpassungen an der Zeitform des zitierten Textes im Kontext des jeweiligen Abschnittes vorgenommen. Auf eine explizite Hervorhebung haben wir deshalb bewusst verzichtet.

- [AM] Projektseite von Automapper. Adresse: <https://automapper.org/> (besucht am 4. Juni 2019).
- [Angular] Webseite des *Angular* Projekts. Adresse: <https://angular.io/> (besucht am 1. Juni 2019).
- [Angular-Arch] Übersicht der Architektur von Angular Applikationen. Adresse: <https://angular.io/guide/architecture> (besucht am 7. Juni 2019).
- [angular-oauth2-oidc] angular-oauth2-oidc Github Repository. Adresse: <https://github.com/manfredsteyer/angular-oauth2-oidc> (besucht am 10. Juni 2019).
- [ASPNET-Tutorial] Einführung in API Entwicklung mit ASP.NET Core. Adresse: <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-2.2&tabs=visual-studio> (besucht am 10. Juni 2019).
- [Clarity] Webseite des *Clarity Design Systems*. Adresse: <https://clarity.design/> (besucht am 1. Juni 2019).
- [Clarity-Intro] Screenshot der drei Grundpfeiler von [\[Clarity\]](#). Adresse: <https://clarity.design/> (besucht am 1. Juni 2019).
- [EF-1] Webseite von Code First Tutorial. Adresse: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/creating-an-entity-framework-data-model-for-an-asp-net-mvc-application> (besucht am 4. Juni 2019).
- [EF-2] Entity Framework Core Dokumentation. Adresse: <https://docs.microsoft.com/en-us/ef/core/managing-schemas/migrations/> (besucht am 4. Juni 2019).
- [ERD] ERD Github Repository. Tool zum erstellen von Entity Relationship Diagrammen aus textuellen Beschreibungen. Adresse: <https://github.com/BurntSushi/erd> (besucht am 7. Juni 2019).
- [Gitstats] Git Statistik Tool. Adresse: <http://gitstats.sourceforge.net/> (besucht am 11. Juni 2019).

- [IM] Definition des Begriffs *Identity Management*. Adresse: <https://searchsecurity.techtarget.com/definition/identity-management-ID-management> (besucht am 9. Juni 2019).
- [Jazz-OIDC-NET-Core] Proof of Concept einer Integration des *Jazz Authorization Server* für *.NET Core*. Adresse: <https://github.com/innerjoin/jazz-oidc-aspnetcore-app> (besucht am 22. April 2019).
- [JWT] JSON Web Token, "industry standard RFC 7519 method for representing claims securely between two parties". Adresse: <https://jwt.io/> (besucht am 7. Juni 2019).
- [load-balancing] Nginx Dokumentation zu Load Balancing. Adresse: [http://nginx.org/en/docs/http/load\\_balancing.html](http://nginx.org/en/docs/http/load_balancing.html) (besucht am 9. Juni 2019).
- [Madge] Madge Github Repository. Tool zum erstellen von Abhängigkeitsgraphen von Javascript. Adresse: <https://github.com/pahen/madge> (besucht am 7. Juni 2019).
- [MS-SQL-Temporal-Tables] Dokumentation der Temporal Tables Funktion von MS SQL. Adresse: <https://docs.microsoft.com/en-us/sql/relational-databases/tables/temporal-tables?view=sql-server-2016> (besucht am 1. Juni 2019).
- [MTA-einfach-erklärt] Die Meilensteintrendanalyse einfach erklärt, Andrea Windolph & Dr. Alexander Blumenau. Adresse: <https://projekte-leicht-gemacht.de/blog/pm-methoden-erklärt/die-meilensteintrendanalyse-einfach-erklärt/> (besucht am 6. Juni 2019).
- [NET-CORE-API] Überblick von ASP.NET Core API Architektur. Adresse: <https://dotnet.microsoft.com/apps/aspnet/apis> (besucht am 7. Juni 2019).
- [NSwag] Github Projekt von NSwag. Adresse: <https://github.com/RicoSuter/NSwag> (besucht am 4. Juni 2019).
- [oauth2] RFC von OAuth 2.0, Adresse: <https://tools.ietf.org/html/rfc6749> (besucht am 10. Juni 2019).
- [oidc-flow] Dokumentation des OpenID Connect 1.0 Implicit Flow, Adresse: [https://openid.net/specs/openid-connect-core-1\\_0.html#ImplicitFlowAuth](https://openid.net/specs/openid-connect-core-1_0.html#ImplicitFlowAuth) (besucht am 10. Juni 2019).
- [OpenAPI] OpenAPI Spezifikation. Adresse: <https://swagger.io/docs/specification/about/> (besucht am 4. Juni 2019).
- [Pixabay-1] Bild zur Persona *Praveen*, Nr. 210382. Adresse: <https://pixabay.com/de/photos/210382/> (besucht am 2. Juni 2019).
- [Pixabay-2] Bild zur Persona *Quentin*, Nr. 1281282. Adresse: <https://pixabay.com/de/photos/1281282/> (besucht am 2. Juni 2019).

- [Pixabay-3] Bild zur Persona *Monera*, Nr. 2773007. Adresse: <https://pixabay.com/de/photos/2773007/> (besucht am 2. Juni 2019).
- [Pixabay-4] Bild zur Persona *Olivia*, Nr. 1280966. Adresse: <https://pixabay.com/de/photos/1280966/> (besucht am 2. Juni 2019).
- [Pixabay-5] Bild zur Persona *Colin*, Nr. 2194255. Adresse: <https://pixabay.com/de/photos/2194255/> (besucht am 2. Juni 2019).
- [PP] A. Hunt and D. Thomas, "The Evils of Duplication" in *The Pragmatic Programmer, 20th Anniversary Edition*, Pearson/Addison-Wesley, 2019. [E-book]
- [SA] L. Steiger und P. Hürlimann, Studienarbeit (SA) "*NextGen Project Management Tooling*", HSR Hochschule für Technik Rapperswil, Dez. 2018.
- [SA-1] , [SA] - "Management Summary" → "Aufgabenstellung", Seite 1.
- [SA-2] , [SA] - "Technologiestack" → 3.1.2, Seite 41.
- [SA-3] , [SA] - "Frontend Framework & UI Library" → 3.3.1, Seite 52.
- [SA-4] , [SA] - "Anforderungsanalyse" → 2.1.3, Seite 13.
- [SA-5] , [SA] - "Continuous Integration and Delivery" → 3.6.1, Seite 76.
- [SA-6] , [SA] - "Containerization" → 3.3.5, Seite 60.
- [SA-7] , [SA] - "Lösungsstrategie" → 3.3, Seite 48.
- [SA-8] , [SA] - "Verteilungssicht" → 3.6, Seite 77.
- [SA-9] , [SA] - "User Experience" → 3.7.1, Seite 81.
- [SA-10] , [SA] - "Architektur" → 3.7.2, Seite 79.
- [SA-11] , [SA] - "Identity Management" → 3.1.1, Seite 43.
- [SA-12] , [SA] - "Ausblick" → 4.2, Seite 95.
- [SA-21] , [SA] - "Anforderungsanalyse" → 2.1. Einführung und Ziele, zu finden als Auszug im Anhang unter [Ziele & Stakeholder](#).
- [SA-22] , [SA] - "Anforderungsanalyse" → 2.2. Nutzungsanalyse, Seite 15, Seite 10, zu finden als Auszug im Anhang unter [Nutzungsanalyse](#).
- [SA-23] , [SA] - "Anforderungsanalyse" → 2.3. Bestandsanalyse, Seite 20, zu finden als Auszug im Anhang unter [Bestandsanalyse](#).
- [SA-24] , [SA] - "Anforderungsanalyse" → 2.4. Personas, Seite 29, zu finden als Auszug im Anhang unter [Personas](#).

- [SA-25] , [SA] - "Anforderungsanalyse" → 2.5. Benutzungsszenarien, Seite 35, zu finden als Auszug im Anhang unter [Benutzungsszenarien](#).
- [SA-35] , [SA] - "3.5.3. Authentication & Authorization Backend" → 3.5.3, Seite 74.
- [SAFe-ART] Erläuterung eines Agile Release Train als Teil des Scaled Agile Framework. Adresse: <https://www.scaledagileframework.com/agile-release-train/> (besucht am 1. Juni 2019).
- [Schaefer18] Mündliche Aussage während den Anwenderinterviews im Rahmen unserer Studienarbeit, Lothar Schaefer, Project Quality Manager bei [Siemens](#).
- [Schneider19] Mündliche Aussage während Zwischenpräsentation der Bachelorarbeit, Guido Schneider, Head "Tools & Infrastructure" bei [Siemens](#).
- [Siemens] Der Term *Siemens* wird im Rahmen dieser Arbeit für unseren Auftraggeber genutzt. Aus rechtlicher Sicht war und ist das die *Siemens Schweiz AG* mit Sitz in Zug. Aus organisatorischer Sicht war zum Startzeitpunkt der Arbeit unser Auftraggeber aus der Division *Building Technologies* mit ihrem globalen Hauptsitz in Zug. Aufgrund von organisatorischen Änderungen sind diese nun ein Teil der neu geschaffenen *Operating Company Smart Infrastructure* unter dem Namen *Building Products*. Um Konfusion zu vermeiden, wird ausschliesslich der relativ allgemein gehaltene Name "Siemens" verwendet.
- [silent-refresh] Erklärung zu Silent Refresh für Oidc Tokens. Adresse: <https://www.scottbrady91.com/OpenID-Connect/Silent-Refresh-Refreshing-Access-Tokens-when-using-the-Implicit-Flow> (besucht am 10. Juni 2019).
- [SOAP] Spezifikation des SOAP Protokolls. Adresse: <https://www.w3.org/TR/soap/> (besucht am 1. Juni 2019).
- [Swagger-API] REST API der während der Bachelorarbeit entwickelten Software. Adresse: <https://sinv-56083.edu.hsr.ch/swagger/index.html> (besucht am 7. Juni 2019).
- [Wiki-ETL] Artikel zu ETL-Prozessen. Adresse: <https://de.wikipedia.org/wiki/ETL-Prozess/> (besucht am 1. Juni 2019).
- [Wiki-MTA] Artikel zur Meilensteintrendanalyse. Adresse: <https://de.wikipedia.org/wiki/Meilensteintrendanalyse/> (besucht am 1. Juni 2019).

## 8. Anhang

### 8.1. Persönliches Fazit

Wenn wir gemeinsam auf die zurückliegende Studien- und Bachelorarbeit zurückblicken, können wir mit gutem Gewissen sagen, dass wir mit der Umsetzung und dem Resultat sehr zufrieden sind.

Ohne die tatkräftige Unterstützung vieler Personen wäre diese Arbeit nicht zu diesem für uns mehr als befriedigenden Ergebnis gekommen. Deshalb möchten wir an dieser Stelle die Möglichkeit nutzen, um folgenden Personen unseren aufrichtigen Dank auszusprechen.

- **Daniel Keller** für sein kontinuierliches, aufrichtiges, konstruktives Feedback, welches er uns Woche für Woche gegeben hat.
- Unseren **Teamkollegen** bei Siemens, welche uns mit Rat und Tat zur Seite gestanden sind. Insbesondere **Thomas Weh** für die Ausschreibung und Finanzierung dieser Arbeit.
- **Peter Heinzmann** und **Hans Rudin** für ihre wertvolle Aussenperspektive, welche sie zur Mitte der Bachelorarbeit eingebracht haben und Licht in unseren "*Tunnelblick*" gebracht haben.
- **Freunden und Familie** für die teilweise zu kurz gekommene Zeit während der Arbeit.

Wie wir beide als Individuum die Arbeit erlebt haben, kann den beiden folgenden persönlichen Selbstreflexionen entnommen werden.

#### 8.1.1. Pascal Hürlimann

Wir haben uns sowohl für die Studien- wie auch die Bachelorarbeit ambitionierte Projekte vorgenommen. Ich war mir zu Beginn nicht sicher, ob wir es wirklich schaffen würden einen Weg zu finden aus der unüberschaubar wirkenden Umgebung einen Teil so heraus zu lösen wie wir es bewerkstelligt haben. Die Chance über die Zeit beider Arbeiten ein Projekt zu verfolgen hat uns die Möglichkeit gegeben über längere Zeit nachvollziehen zu können, welche organisatorischen und technischen Entscheidungen erfolgreich waren, und welche nicht. Wir konnten über die Dauer beider Projekte unsere jeweiligen Stärken ausspielen und haben uns dabei immer gut ergänzt. Ernüchternd, aber nicht unerwartet, war es wieder ein Mal zu sehen, wie äussere Faktoren ein Projekt ausbremsen können. Ich freue mich, dass wir beide um diese Erfahrung reicher geworden sind und unsere Problemlösungsfähigkeiten unter Beweisen stellen durften.

### 8.1.2. Lukas Steiger

Um ein Projekt erfolgreich durchführen zu können, braucht es ein Team, welches harmonisiert. Da Pascal und ich uns bereits zuvor als Arbeitskollegen kennen und schätzen gelernt haben (auch wenn wir kaum gemeinsam an Projekten gearbeitet haben), wusste ich, dass wir uns gut verstehen und ergänzen würden. Da ich während meiner täglichen Arbeit wenig im Team entwickle und kaum an umfangreicheren Projekten arbeite, wollte ich unbedingt ein realitätsnahes, umfangreicheres Projekt als Arbeit umsetzen, um auch mal möglichst am gesamten Entwicklungszyklus eines Projektes entscheidend mitwirken zu können. Die nun zusammen mit Pascal umgesetzte Studien- und Bachelorarbeit konnte meine Erwartungen diesbezüglich voll erfüllen, was mich wirklich freut. Sehr hilfreich und entscheidend für den Erfolg war, dass wir stets zeitnahe, konstruktives und zielführendes Feedback von unseren Kollegen bei Siemens erhalten haben. Gleiches gilt auch für die zahlreichen Gespräche, Rückmeldungen und Verbesserungsvorschläge von Herrn Keller, welcher uns mit seinem riesigen Erfahrungsschatz stets zur Seite gestanden ist.

## 8.2. Metriken

### 8.2.1. Code Analyse

Die Code Analyse bezieht sich auf Dateien, welche den Umfang des Projekts einigermaßen widerspiegeln. Die Statistiken beruhen auf Daten gesammelt von Github und aus [\[Gitstats\]](#).

Tabelle 5. Anzahl Zeilen Source Code (Quelle: Git Stats)

Bereich	Anzahl Dateien	Anzahl Zeilen Code
Backend (C#)	80	2285
Frontend Scripts (TypeScript, JS)	88	2552
Frontend Darstellung (HTML, CSS, Templates)	61	2039
SSIS + T-SQL (~80% generierter Code)	12	16032
Konfigurationsdateien	16	314
Dokumentation	47	4058

### 8.2.2. Commit Analyse

Die mit [\[Gitstats\]](#) erstellte Commit Analyse zeigt die Anzahl Commits pro Student auf. Diese verleiht ein gutes Bild der stets fairen und gleichmässigen Arbeitsaufteilung. Gleichzeitig sieht man, dass wir es geschafft haben die Arbeit gleichmässig auf die Iterationen einzuteilen.

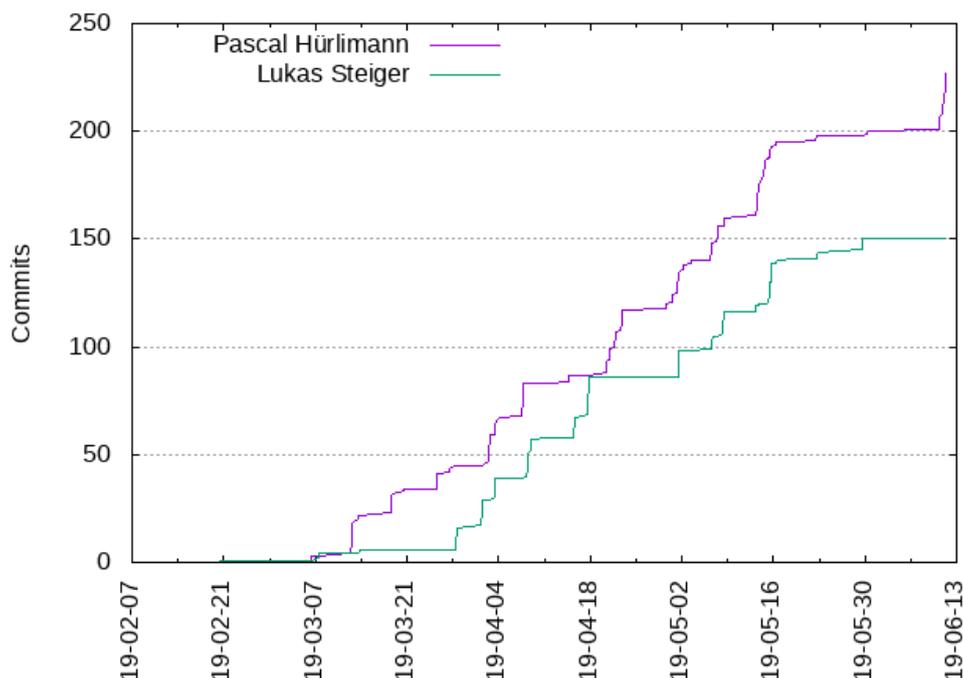


Abbildung 63. Commit Analyse des Code Repositories

### 8.2.3. Zeitauswertung

Auswertung des Zeitaufwandes, je ein Mal aufgeschlüsselt pro Kategorie, Sprint und Student.

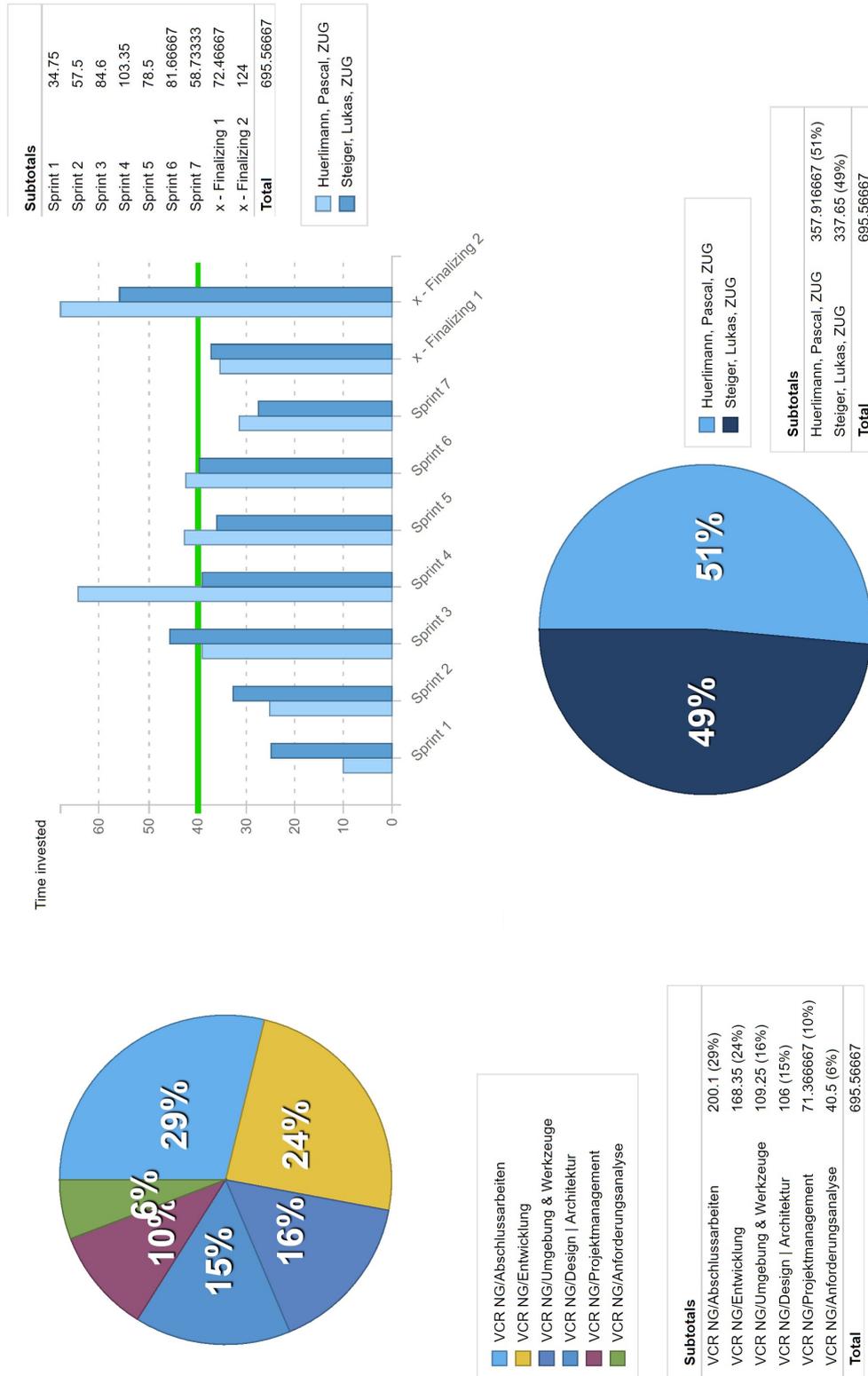
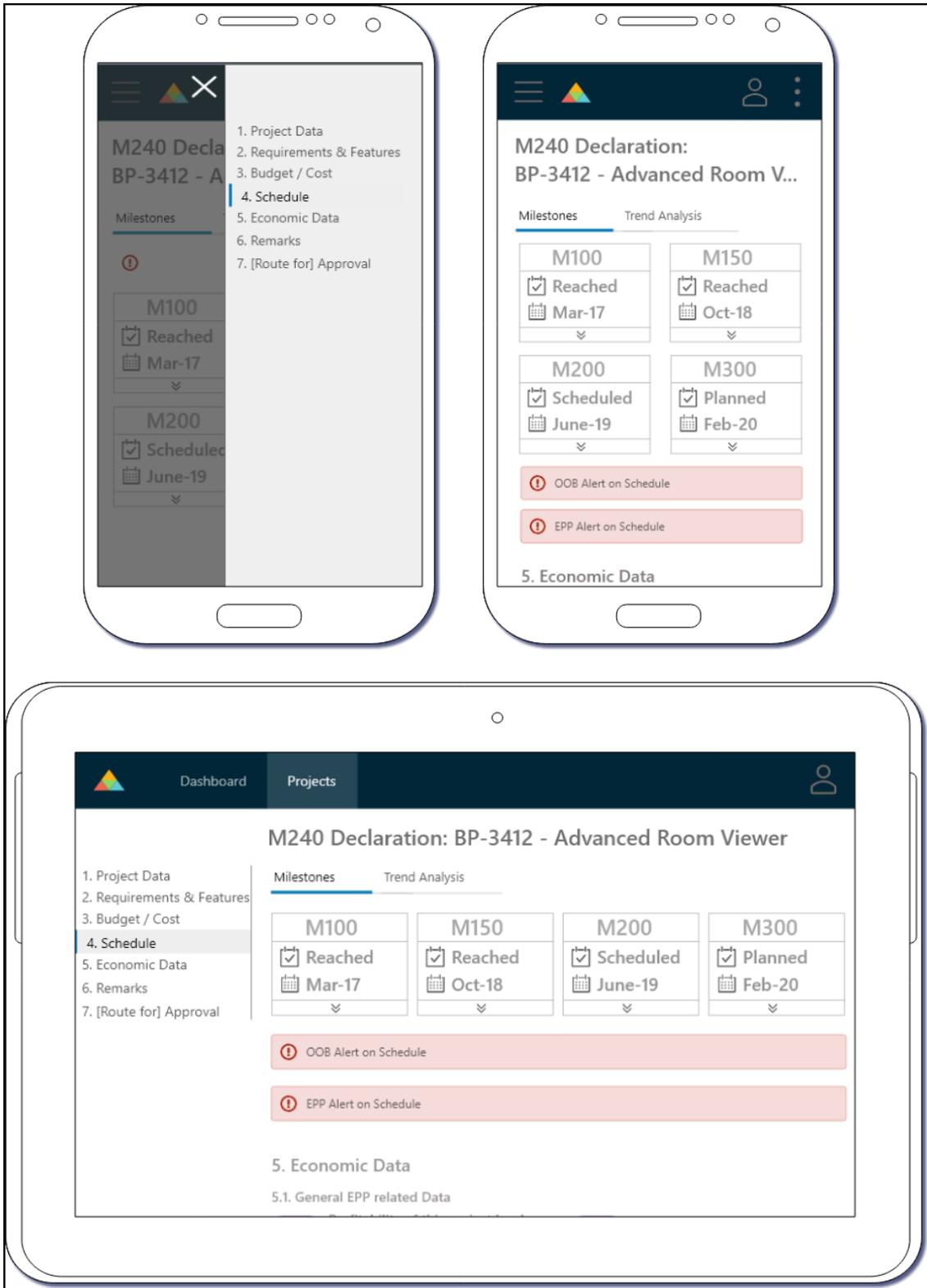


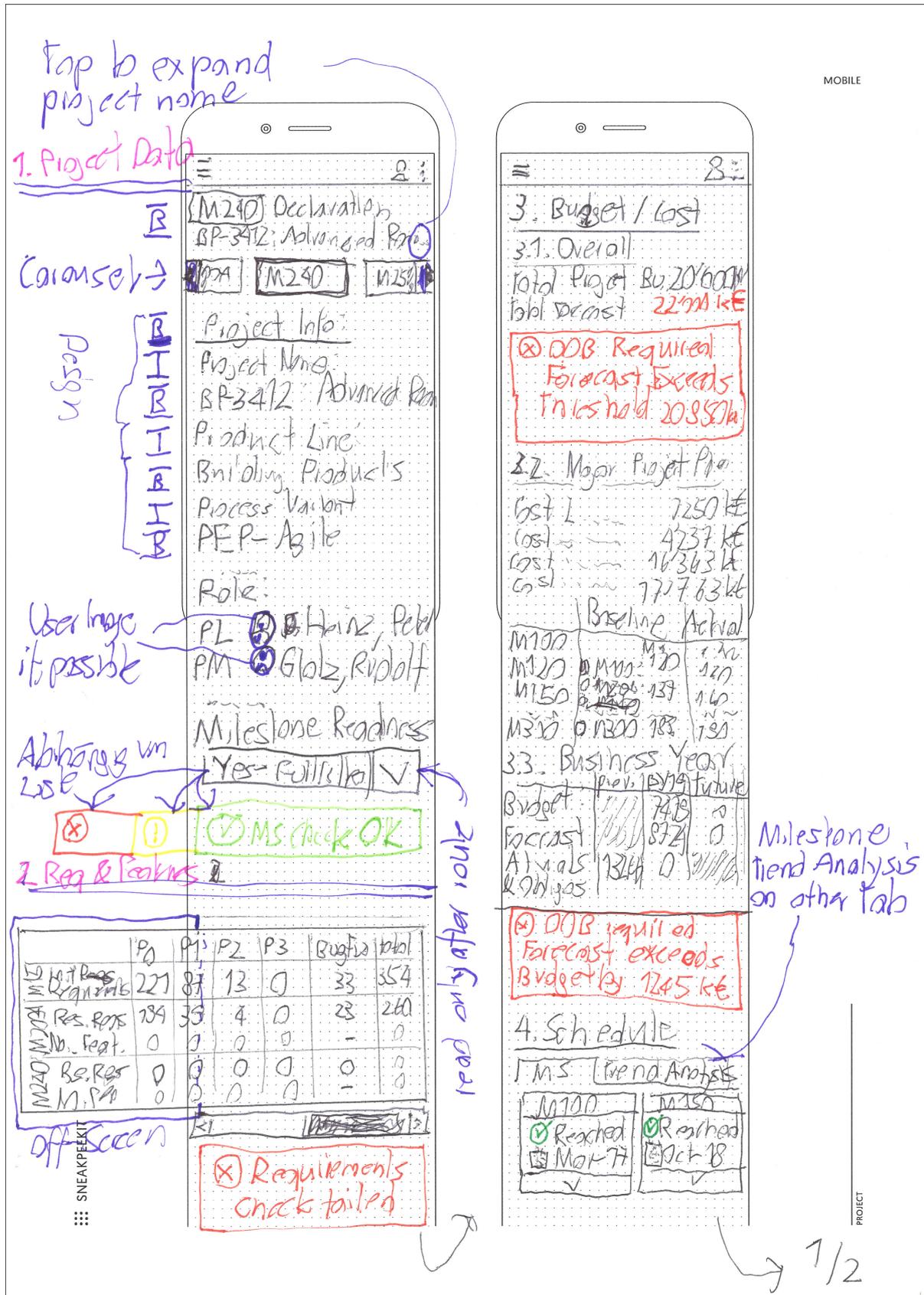
Abbildung 64. Zeitauswertung aus IBM Report Builder

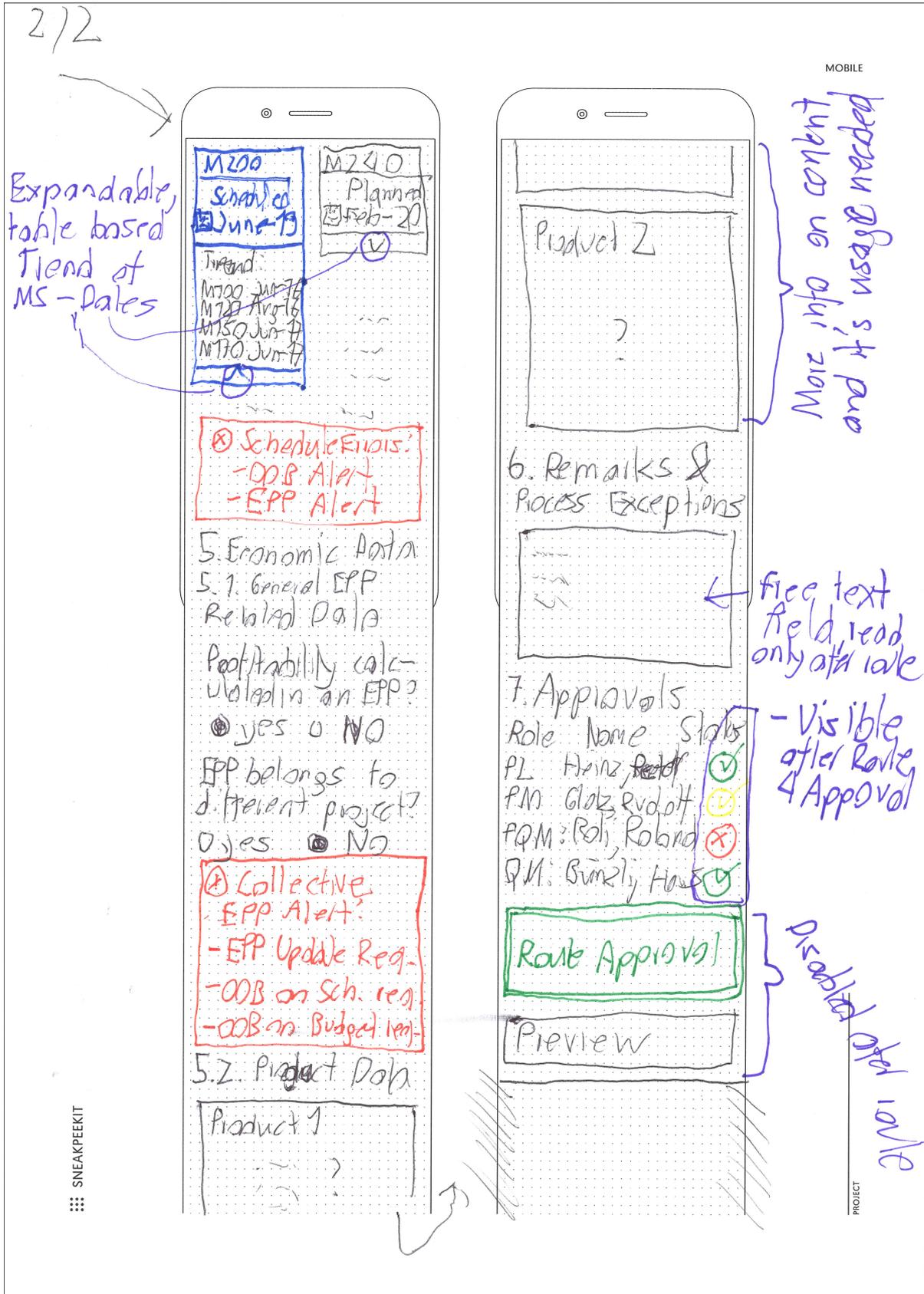
## 8.3. Prototypen & Wireframes

### 8.3.1. Digitaler Prototyp



8.3.2. Detaillierte Wireframes auf Papier





## 8.4. Evaluationen

Während der Bachelorarbeit wurden die nachfolgend erläuterten Evaluationen durchgeführt. Die bereits während der Studienarbeit durchgeführten Evaluationen, z.B. für ein *Frontend Framework*, wurden komplett weggelassen.

### 8.4.1. ETL Software Evaluation

Der Einsatz von ETL Jobs ist eine verbreitete und etablierte Methode, um Daten aus verschiedenen Datenquellen zu aggregieren und in eine konforme Struktur zu bringen. Daher ist es nicht erstaunlich, dass es eine Vielzahl von technischen Lösungen gibt, welche zuerst evaluiert werden mussten. Es gibt bei Siemens keine weitläufig etablierte ETL Lösung, womit die Wahl eines passenden Tools grösstenteils den Studenten überlassen wurde. Bevorzugt wird, wenn die gewählte Lösung keine zusätzlichen Lizenzkosten erfordert und sich gleichzeitig so nahtlos wie möglich in die bestehende Infrastruktur einfügt.

Da keine weiteren Anforderungen an diese Lösung gestellt wurden, soll die gewählte Technologie möglichst optimal den Datenaustausch zwischen den beiden Tools gewährleisten. Dementsprechend wurde nach einer Lösung gesucht, welche Microsoft SQL Server verwenden lässt und mit der die Transformation relationaler Daten besonders einfach ist. Auch die Einarbeitungszeit für das Team ist kein unerheblicher Faktor, da die Zeit ein stark einschränkender Faktor der Bachelorarbeit darstellt.

### Open Source Lösungen

Um der Anforderung der Lizenzkosten nachzukommen, wurde sich zuerst ein Überblick über existierende Open Source Lösungen verschafft. Zwei führende und seit Jahren etablierte Lösungen mit Open Source Lizenzen sind Pentaho Data Integration (auch genannt Kettle) und Talend Open Studio. Beide basieren auf kommerziellen Produkten, von welchen eingeschränkte Versionen der Community als Open Source zur Verfügung gestellt werden. Mit Apache NiFi ist eine weitere Lösung in die nähere Auswahl aufgenommen worden, welche ebenfalls mit einer Open Source Lizenz angeboten wird.

### Pentaho Data Integration

Pentaho als Produktfamilie bietet für diverse datenorientierte Workflows Lösungen. Die Data Integration Komponente befasst sich mit dem Entwerfen und Ausführen von Datenintegrations- und Transformationsworkflows.

Vorteile:

- Grafische Designoberfläche.
- Volle Integration mit der gesamten Pentaho Plattform.
- Möglichkeit, später Enterprise Support zu erhalten.

Nachteile:

- Zusätzliche Komponente mit wenig Integration in aktuelles Deployment der Lösung.
- Dem Team unbekannte Software, daher hoher Aufwand zur Einarbeitung.
- Wenig Flexibilität bei der Anpassung von generierten Komponenten.
- Unübersichtliches proprietäres Angebot.

### Talend Open Studio

Talend Open Studio ist eine ETL Lösung der Firma Talend, welche auch weitere Daten orientierte Softwarelösungen entwickelt. Viele Dienste werden sowohl als deploybare Pakete wie auch Cloud-Services angeboten.

Vorteile:

- Eclipse basierte Designoberfläche, das Team ist sehr vertraut mit der Eclipse Plattform.
- Generiert ausführbare Javabinaries, daraus ergibt sich eine einfache Automation mit Jenkins.
- Möglichkeit, später Enterprise Support zu erhalten.

Nachteile:

- Teures, Userbasiertes Abonnementmodell.
- Open Source Version scheint die eingeschränkteste der Auswahl zu sein.

### Apache NiFi

Apache NiFi ist eine Open Source ETL Lösung welche von Hortonworks entwickelt und unterstützt wird. Die Entwicklung von ETL Jobs basiert auf einer flowbasierten, grafischen, Graphbeschreibungssprache.

Vorteile:

- Sehr aktive Community.
- Gesamte Plattform ist Open Source.

- Flexible Gestaltung der Data-flows.

Nachteile:

- Ausschliesslich Flow-based (grafisches) Programmieren.
- Weitere Komponente fürs Deployment nötig.
- Eingeschränkter Support für NiFi in Vergleich zu restlicher Plattform.

## Proprietäre Lösungen

### SQL Server Integration Services

SQL Server Integration Services (SSIS) ist eine ETL Lösung von Microsoft, welche vorwiegend für die Integration mit anderen Microsoft Lösungen entwickelt wird. SSIS ist eine sehr flexible Lösung zur Entwicklung von ETL Jobs, deren Designoberfläche in Visual Studio integriert ist.

Vorteile:

- Hervorragende Integration mit der bestehenden Plattform. Der SSIS Dienst ist Teil von SQL Server und muss nur aktiviert werden.
- Sehr flexible Entwicklungsumgebung. Dataflows können grafisch, in SQL oder auch in gängigen .NET Programmiersprachen erstellt werden.
- Support als Teil von SQL Server bereits abgedeckt, die Entwicklung von Paketen ist mit einer Gratislizenz erlaubt.

Nachteile:

- Plattformeinschränkungen. Zur Zeit dieser Arbeit können SSIS Pakete nur auf native Instanzen von SQL Server deployt werden.
- Unflexibelste Lösung, das Hauptaugenmerk liegt auf der Integration mit Microsoftprodukten.

## Fazit

Es gibt ein umfangreiches Ökosystem von Datenintegrationstools, mit unterschiedlichen Ansätzen zur Entwicklung von ETL Jobs. Verschiedene Lösungen bieten andere Stärken und Schwächen, wobei es nicht unerheblich ist zu betrachten, in welchem Kontext diese eingesetzt werden sollen.

Durch die einzigartig gute Integration mit der bereits bestehenden Infrastruktur, ist die Wahl

schnell auf SSIS als ETL Lösung gefallen. Mit SSIS ist es allein durch die Aktivierung eines zusätzlichen Dienstes möglich, ETL Jobs in die Produktivumgebung zu integrieren. Für die angestrebte Lösung werden die umfangreicheren Features anderer Tools nicht benötigt, dafür erfordern Inbetriebnahme und Wartung von SSIS bedeutend weniger Aufwand.

## 8.5. Einsatz von Third Party Software

Unsere Software basiert auf verschiedenen Softwarekomponenten von Drittherstellern. Zum Ende dieser Arbeit wurden die eingesetzten Softwareversionen auf den neusten Stand gebracht und eine Analyse der eingesetzten Software durchgeführt.

Die Durchführung ist am **29. Mai 2019** erfolgt. Sämtliche Angaben waren zu diesem Zeitpunkt aktuell, können sich aber zum Lesezeitpunkt davon unterscheiden.

### 8.5.1. Eingesetzte Software von Drittanbietern

Die nachfolgenden Tabellen erläutern sämtliche im Endprodukt eingesetzten Drittherstellerkomponenten auf.

#### Backend

*Tabelle 6. Auflistung sämtlicher im Backend direkt genutzten Softwarekomponenten von Drittanbietern*

Komponente	Version	Lizenz	Verwendungszweck	Bemerkungen
AutoMapper	8.1	MIT	Automatisches Mapping zwischen den von EF genutzten Model Klassen und den DTO's, welche über das API zur Verfügung gestellt werden.	
AutoMapper .Collection	5.0	MIT	AutoMapper Unterstützung für die Aktualisierung bestehender <i>Collections</i> .	
Microsoft.AspNet-Core.App	2.2.5	Apache License 2.0	ASP.NET für .NET Core.	
Microsoft.AspNet-Core.Authentication.JwtBearer	2.2.0	Apache License 2.0	Austausch von <i>JSON Web Tokens</i> zwischen <i>Keycloak</i> und unserem Backend.	

Komponente	Version	Lizenz	Verwendungszweck	Bemerkungen
Microsoft .CodeAnalysis .FxCopAnalyzers	2.6.2	Apache License 2.0	Source Code Analysetool für C#. Verletzungen von Codierungsrichtlinien resultieren in <i>Build Warnings</i> .	Nutzung einer älteren Version (2.6.2 statt 2.9.2). Neuere Versionen weisen den folgenden Bug auf: <a href="https://github.com/AutoMapper/AutoMapper/issues/3053">https://github.com/ AutoMapper/AutoMapper/ issues/3053</a>
Microsoft.Entity- FrameworkCore.*	2.2.4	Apache License 2.0	Framework für objektrelationale Datenbankanbindung. Beschreibung des Datenbankmodells mit annotierten C# Model Klassen	
Microsoft.NETCore .App	2.2.0	MIT	Eine Reihe von .NET-APIs, die im Standard- Anwendungsmodell.NET Core enthalten sind.	
NSwag.*	12.3.1	MIT	Swagger für .NET. Wird genutzt, um automatisch aus den annotierten C# Controllern eine Swagger Spezifikation sowie clientseitiges Typescript zu erzeugen.	
StyleCop .Analyzers	12.3.1	MIT	Implementierung der StyleCop-Regeln zur Prüfung der Einhaltung der Codierrichtlinien.	

## Frontend

Tabelle 7. Auflistung sämtlicher im Frontend direkt genutzten Softwarekomponenten von Drittanbietern

Komponente	Version	Lizenz	Verwendungszweck	Bemerkungen
@angular/*	7.2.15	MIT	Angular Framework - Frontend Framework	Neue Version '8.0' wurde gestern, 28.5., veröffentlicht. Upgrade noch nicht möglich, u.a. wegen VMWare Clarity, wo Version 2.0 abgewartet werden muss.
@clr/*	1.1.3	MIT	VMWare Clarity - UI/UX Framework mit eingebauter Angular Unterstützung	Fonts are licended under the SIL Open Font License version 1.1
@ctrl/ngx-headroom	2.3.2	MIT	Scrolling navigation header	
@webcomponents/custom-elements	1.2.4	BSD 3 Clause	HTML Custom Elements Polyfill	
angular-oauth2-oidc	5.0.2	MIT	Support for OAuth 2 and OpenId Connect (OIDC) in Angular	
core-js	3.1.3	MIT	Javascript core Library	
moment	2.24.0	MIT	Date Library	
rxjs	6.5.2	Apache 2.0	Reactive Extensions for modern JavaScript	
zone.js	0.8.26	MIT	Zones for JavaScript - required by Angular	Newer version 0.9.1 available. Currently installed version is a required peer dependency of Angular 7.x

## 8.5.2. Prüfung auf bekannte Sicherheitslücken

Die aktuell eingesetzten Komponenten wurden zum oben genannten Zeitpunkt auf bekannte Sicherheitslücken, sogenannte *Common Vulnerabilities and Exposures*, geprüft.

### Backend

Im Backend gibt es aktuell eine potentielle Sicherheitslücke in der Komponente `System.Net.Http - 4.3.0`. Diese Komponente wird jedoch nicht direkt von uns eingebunden, sondern via `Microsoft.IdentityModel.Protocols - 5.3.0`, welches wiederum ein Teil von `Microsoft.AspNetCore.App - 2.2.5` ist.

Das Analyseergebnis von `dotnet retire` sieht so aus:

```
1 info: dotnet_retire.RetireLogger[0]
2     [32mScan starting [0m [39m [49m
3 fail: dotnet_retire.RetireLogger[0]
4     Found use of 1 vulnerable libs in 3 dependency paths. [31m
5
6 * System.Net.Http/4.3.0 [0m [39m [49m
7
8 info: dotnet_retire.RetireLogger[0]
9     Scan complete.
```

### Frontend

Im Frontend konnten alle potentiellen Sicherheitslücken durch Aktualisierungen auf aktuellste Versionen behoben werden. Die Überprüfung mit `npm audit` hat folgendes Ergebnis gebracht:

```
1 === npm audit security report ===
2
3 found 0 vulnerabilities
4 in 42885 scanned packages
```

### 8.5.3. Wartung & Audit

Die folgende Sektion beschreibt, wie die Wartung der Applikation im Bezug auf Komponenten von Drittherstellern erfolgen soll.

#### Backend

##### Upgrade von Komponenten

Für die Wartung des Backend eignet sich `dotnet-outdated`. Die Installation kann wie folgt vorgenommen werden:

```
dotnet tool install --global dotnet-outdated
```

Damit lassen sich neuere Versionen für bereits installierte Pakete detektieren:

```
dotnet outdated -i
```

Die Analyse bringt Resultate im folgenden Stil:

```
1 > dotnet outdated -i
2
3 Backend
4 [ .NETCoreApp,Version=v2.2 ]
5 Microsoft.AspNetCore.App [A]          2.2.0 -> 2.2.5
6 Microsoft.CodeAnalysis.FxCopAnalyzers 2.6.2 -> 2.9.2
7 Microsoft.NETCore.App [A]            2.2.0 -> 2.2.5
```

Ein manuelles Upgrade, Paket für Paket, kann mit folgendem Kommando vorgenommen werden:

```
dotnet outdated -i -u:prompt
```

##### Prüfung auf Sicherheitslücken

Für die Prüfung auf bekannte Sicherheitslücken im Backend eignet sich `dotnet-retire`. Die Installation kann wie folgt vorgenommen werden:

```
dotnet tool install --global dotnet-retire
```

Die Prüfung auf bekannte Sicherheitslücken kann wie folgt durchgeführt werden:

```
dotnet retire
```

Die Resultate werden wie folgt zurückgegeben:

```
1 >dotnet retire
2
3 info: dotnet_retire.RetireLogger[0]
4     [32mScan starting [0m [39m [49m
5 fail: dotnet_retire.RetireLogger[0]
6     Found use of 1 vulnerable libs in 3 dependency paths. [31m
7
8 * System.Net.Http/4.3.0 [0m [39m [49m
9
10 info: dotnet_retire.RetireLogger[0]
11     Scan complete.
```

Für genauere Informationen, insbesondere welche verschachtelte Abhängigkeit eine Sicherheitswarnung auslöst, kann mit dem `loglevel` gearbeitet werden:

```
1 > dotnet retire loglevel=debug
2 ....
3
4 * System.Net.Http/4.3.0 [0m [39m [49m
5
6     Microsoft.AspNetCore.App/2.2.5
7     └─ Microsoft.AspNetCore.Authentication.WsFederation/2.2.0
8     └─ Microsoft.IdentityModel.Protocols.WsFederation/5.3.0
9         └─ Microsoft.IdentityModel.Protocols/5.3.0
10             └─ System.Net.Http/4.3.0
11
12 ....
13
14 info: dotnet_retire.RetireLogger[0]
15     Scan complete.
```

## Frontend

### Upgrade von Komponenten

Für die Wartung des Frontend eignet sich `npm-check`. Die Installation kann wie folgt vorgenommen werden:

```
npm install -g npm-check
```

Damit lassen sich neuere Versionen für bereits installierte Pakete detektieren:

```
npm-check -u
```

Die Analyse bringt Resultate im folgenden Stil:

```
1 > npm-check -u
2
3 ? Choose which packages to update. (Press <space> to select)
4
5 Patch Update Backwards-compatible bug fixes.
6 ( ) @angular/http          7.2.8      7.2.15
https://github.com/angular/angular#readme
7 ....
8
9 Minor Update New backwards-compatible features.
10 ( ) rxjs                   6.4.0      6.5.2   https://github.com/ReactiveX/RxJS
11 ....
12
13 Major Update Potentially breaking API changes. Use caution.
14 ( ) typescript devDep     2.1.4      3.4.5   https://www.typescriptlang.org/
15 ....
```

Ein manuelles Upgrade, Paket für Paket, kann mit folgendem Kommando vorgenommen werden:

```
dotnet outdated -i -u:prompt
```

### Lizenzprüfung

Lizenzinformationen können mit `legally` geprüft werden. Installation via

```
npm i legally -g
```

Lizenzprüfung für ein bestimmtes Paket:

```
legally <package>
```

### Prüfung auf Sicherheitslücken

Die `audit` Funktion von NPM ist hier geeignet:

```
npm audit
```

Mit dem Kommando `npm audit fix` lassen sich einige davon automatisch beheben, indem im `package-lock.json` die entsprechenden Minimalversionen aktualisiert werden.

Ein `npm audit` Beispiel sieht wie folgt aus:

```
1  ....
2
3  High          Prototype Pollution
4  Package       handlebars
5  Dependency of karma-coverage-istanbul-reporter [dev]
6  Path          karma-coverage-istanbul-reporter > istanbul-api >
7               istanbul-reports > handlebars
8  More info     https://nodesecurity.io/advisories/755
9
10 found 5 vulnerabilities (2 moderate, 3 high) in 42883 scanned packages
11 run `npm audit fix` to fix 5 of them.
```

## 8.6. SQL Code Ausschnitt

*Beispiel einer optimierten, aber komplexen SQL Abfrage*

```

1 SELECT
2   project.id           AS 'ProjectId',
3   report.id           AS 'ReportId',
4   report.reportingdate AS 'ReportingDate',
5   estimate.id        AS 'EstimateId',
6   estimate.reached    AS 'EstimateReached',
7   milestone.id       AS 'MilestoneId',
8   milestone.milestone AS 'Milestone Name',
9   process.id         AS 'ProcessId',
10  process.process     AS 'Process Name',
11  estimate.estationdate,
12  CASE estimate.reached
13  WHEN 1 THEN
14    'Reached'
15  ELSE 'Planned' END   AS 'Status'
16 FROM
17   VCR_HQS_BT_CPS.dbo.t_project AS project
18 INNER JOIN
19   VCR_HQS_BT_CPS.dbo.t_milestonereport AS report
20   ON report.project_id = project.id
21 INNER JOIN
22   VCR_HQS_BT_CPS.dbo.t_milestoneestimation AS estimate
23   ON estimate.milestonereport_id = report.id
24 INNER JOIN
25   VCR_HQS_BT_CPS.dbo.t_milestone AS milestone
26   ON estimate.milestone_id = milestone.id
27 INNER JOIN
28   VCR_HQS_BT_CPS.dbo.t_process AS process
29   ON project.process_id = process.id
30 INNER JOIN
31   (SELECT
32     project_id,
33     MAX(reportingdate) AS latest
34   FROM
35     VCR_HQS_BT_CPS.dbo.t_milestonereport
36   GROUP BY project_id
37   ) AS latestReport
38   ON latestReport.project_id = project.id
39   AND report.reportingdate = latestReport.latest
40 ORDER BY 'ProjectId', 'ReportingDate'

```

## 9. Auszüge aus der Studienarbeit

### WICHTIG

Im nachfolgenden Kapitel finden Sie Auszüge aus der Studienarbeit *"NextGen Project Management Tooling"*. Die Kapitel wurden eins zu eins übernommen, wobei jedoch entdeckte Schreibfehler oder inkorrekte Satzstellungen korrigiert wurden. Ferner wurden einzelne Kapitelüberschriften angepasst und einzelne Bilder weggelassen, um einen optimalen Lesefluss zu gewährleisten.

## 9.1. Ziele & Stakeholder

*Dieses Kapitel ist ein Auszug aus [\[SA-21\]](#)*

In Rahmen der Anforderungsanalyse wurden die Schwächen und Stärken der bestehenden Landschaft analysiert und Massnahmen zur Umsetzung definiert. Die Hauptmassnahme ist die Neuentwicklung einer Lösung zur Überwachung von Projekten. Diese soll den diversen Aspekten gerecht werden, welche Projektleiter, Qualitäts- und Produktmanagement, Betriebswirtschaft und Top-Management an eine solche Lösung haben. Es soll immer dann zum zentralen Einstiegspunkt werden, wenn um die Kosten- und Fortschrittsverfolgung von Projekten dreht. In einigen Aspekten sollen bestehende Konzepte und Ideen aus der heutigen Bestandlösung VCR übernommen werden, in anderen Fällen sind komplette Neukonzipierungen notwendig. Die neue Lösung muss sich in die bestehende Systemlandschaft integrieren und strategisch wichtige Tools wie Jazz und SAP sollen über Schnittstellen angebunden werden. Auch bei den nichtfunktionalen Anforderungen soll es bedeutende Verbesserungen geben im Vergleich zu VCR, insbesondere bei der Benutzbarkeit, Performanz und der Tauglichkeit für Endgeräte mit diversen Bildschirmgrössen.

### 9.1.1. Zielsetzung für eine neue Applikation

Die neue Applikation soll folgende Funktionalität aufweisen:

- Projektverwaltung mit zentralen Metadaten
- Abbildung des Meilenstein Bewilligungsprozess
- Change Requests für Budget und Zeitplan
- Persönliche Dashboards
- Gefilterte und aggregierte Projektsichten
- Zugriff auf Daten über API
- Aufbereitung der Daten für die noch zu evaluierende Reporting Lösung
- Exports zu Excel

Nebst den funktionalen Anforderungen sind folgende Aspekte auch sehr wichtig:

- Die Applikation soll auf diversen Endgeräten (Mobiltelefone, Tablets, Laptops und Desktop PC's) genutzt werden können. Der zur Verfügung stehende Raum soll dabei jeweils optimal ausgenutzt werden.

- Eine flüssige Bedienung ist sehr wichtig, da die Applikation auch von vielen Mitarbeitenden mit Führungsfunktion genutzt wird. Deren Zeit ist knapp und kostbar für die Firma.
- Bei der Wahl von Dritthersteller Softwarekomponenten (Programmiersprachen, Runtimes, Libraries, Datenbanksysteme) sollen Aspekte wie Lizenz- und Betriebskosten, deren Zukunftsträchtigkeit und deren Lizenzierungsmodell genau angeschaut und abgestimmt werden.

### 9.1.2. Qualitätsziele

Um den Anforderungen unserer Stakeholder gerecht zu werden und für zukünftige Veränderungen gerüstet zu sein, erachten wir die folgenden Qualitätsziele für unsere Architektur als zentral:

*Tabelle 8. Qualitätsziele, nach Priorität*

Ziel	Argumentation
Integration	Der Hauptkritikpunkt der heutigen Lösung VCR ist die fehlende Integration mit anderen Tools. Das System soll von Grund auf die verschiedenen integrativen Aspekte unterstützen. Konzepte wie eindeutige URL's, ein umfangreiches API (z.B. Query/Filter Funktionen), feingranulare API Berechtigungen und zielgerichtete Import/Export Möglichkeiten. Auch Patterns wie Adapter, Plug-in/Extension Interface oder Property List könnten sehr hilfreich für die zukünftige Verwendung sein.
Prozessunabhängig	In der heutigen Lösung gibt es mindestens sechs verschiedene und aktuell verwendete Prozesstypen. Diese unterscheiden sich u.a. in den Rollen, den Zyklen und den Meilensteinen. Auch in Zukunft wird es neue Prozesse geben. Die Architektur soll es erlauben, mit minimalem Aufwand neue Prozesstypen einzuführen.
Modularisierung	Die Applikation betrachtet verschiedene Aspekte des Projektmanagements wie Budget, Meilensteinverfolgung, Qualität, uvm. Diese Aspekte sollen so weit als möglich entkoppelt voneinander sein, sodass in Zukunft neue Module hinzugefügt, bestehende angepasst oder obsolete entfernt/deaktiviert oder durch ein Drittsystem ersetzt werden können.

## 9.2. Nutzungsanalyse

*Dieses Kapitel ist ein Auszug aus [\[SA-22\]](#)*

Um ein besseres Gefühl über die heutige Nutzung der verschiedenen Projektmanagement Tools zu bekommen, haben wir nebst einigen Gesprächen im Umfeld unseres Auftraggebers auch Interviews mit Endanwendern aus verschiedenen Nutzergruppen geführt, um ein ganzheitliches Bild über die heutige Nutzung und die Herausforderungen zu bekommen.

Die Auswertung ebendieser Interviews hat aufgezeigt, dass mit VCR eine Vielzahl von Nutzungsmustern abgedeckt sind. Viele Rollen nutzen VCR für verschiedenste Anwendungsfälle, was den Eindruck stärkt, dass sich VCR historisch immer mehr Funktionalität angeeignet hat, für die es nicht ursprünglich entworfen wurde. VCR bietet die Möglichkeit, auf verschiedenen Abstraktionsstufen Projekte und ihre Daten zu betrachten. Dies ermöglicht es, Projekt Leitern, Qualitätsmanagern aber auch Controllern, VCR sinnvoll für ihre Aufgaben einzusetzen.

### 9.2.1. Interviews

Die von uns interviewten Personen nutzen VCR regelmässig für das Verrichten ihrer Arbeit. Die Interviewpartner wurden so ausgesucht, dass eine möglichst breite Usergruppe vertreten ist. Darunter befanden sich ein Quality Manager, ein Projektleiter, der Leiter des Controllings, ein Project Quality Manager und ein Portfolioleonter. Alle nutzen VCR um verschiedene Ziele zu erreichen und sind daher auf verschiedene Features angewiesen. Die detaillierten Ergebnisse der Interviews befinden sich im Anhang, dieser Abschnitt fasst die wesentlichen Ergebnisse zusammen.

Alle interviewten Personen nutzen VCR regelmässig, wobei sich die genutzten Features und analysierten Daten stark unterscheiden. So sind beispielsweise für die Controller insbesondere die in VCR enthaltenen, globalen Finanzdaten wichtig. Sie nutzen VCR vor allem als Datenschnittstelle, und bearbeiten diese Daten ausserhalb des Tools weiter. Für im Projektmanagement involvierte Personen ist die Projektübersicht von VCR eine zentrale Komponente, da sie so leicht Einblick in den Stand ihrer Projekte erhalten können. Über die vorhandenen Detailansichten können sie einen Drilldown vornehmen, um so mehr Details über den Stand ihres Projekts zu erhalten.

Für die PSD Meetings können verschiedene standardisierte Reports generiert werden, welche während des Meetings einen schnellen Einblick in das Projekt erlauben. Viele Nutzer nutzen dieses Feature, da sie sich so nicht um die manuelle Erstellung dieser Reports kümmern

müssen. Einige Anwender verwenden VCR auch während des Meetings, um sich über gewisse Aspekte von Projekten direkt ein eigenes Bild machen zu können. Hier gewährt die Webansicht einen schnellen Einblick um mit einigen Klicks die gewünschten Informationen darzustellen.

Verschiedene von VCR generierte Reports werden genutzt, um Informationen zu sammeln. So werden vor allem die jeweils über Nacht generierten Exports verwendet, um in Excel die Daten genauer zu analysieren und alternative Reports und Dashboards zu erstellen. Allgemein werden viele Daten aus VCR in Excel weiterverarbeitet. In diesem Fall agiert VCR eher als eine Datenschnittstelle, und nicht als eigentliches Werkzeug.

Alle Benutzer verwenden VCR nicht in Isolation. Es ist eines von zahlreichen Werkzeugen welche sie für die Ausübung ihrer Tätigkeit verwenden. Insbesondere die Kombination aus VCR, Jazz und Excel ist weit verbreitet. Sehr viele individuelle Workflows werden ausserhalb von VCR in Excel vorgenommen, um die Daten weiter aufzubereiten.

VCR wird aktuell als konsolidierter Speicher für alle globalen R&D Finanzdaten verwendet. Zu diesem Zweck werden von jedem Standort die lokalen SAP Datenbanken exportiert und manuell in VCR importiert. Dieser Vorgang wird bald durch eine einheitliche SAP Lösung ersetzt, damit die globalen Finanzdaten leichter und zeitnaher verfügbar gemacht werden. Wann diese neue SAP Lösung den Betrieb aufnimmt und wie sich die Arbeitsabläufe damit verändern werden, ist im Moment noch unklar.

### 9.2.2. Probleme

Einige Probleme haben sich wie ein roter Faden durch die User Interviews gezogen. In diesem Kapitel möchten wir auch auf einige wiederkehrenden Probleme der aktuellen Lösung eingehen und schildern, wie diese den Anwender in seiner alltäglichen Arbeit beeinträchtigen. Es sind vor allem diese Kernprobleme, welche eine neue Lösung primär versuchen sollte zu eliminieren, da so die Auswirkung einer Verbesserung am meisten Nutzer erreichen kann.

Vielen dieser Probleme liegt zugrunde, dass VCR historisch gewachsen ist, und viele Funktionalitäten erst vor zu eingebaut wurden. Auch wurden viele Funktionen eingebaut, die von der ursprünglichen Kernidee (Projektübersicht) abweichen. Daraus ergeben sich viele Features, die nur von wenigen Benutzern verwendet werden. Auch werden einige Anwender dazu gezwungen, das Tool für Zwecke zu nutzen, für die es nie konzipiert wurde. Hier ist vor allem die Finanzplanung und Budgetüberwachung durch das Controlling zu nennen. VCR wurde nicht als Finanztool geplant, sondern diese Funktionalität ist allmählich in das Tool hineingewachsen. So ist der Nutzen insgesamt tief, und das Controlling verlässt sich auf individuell erstellte Excelreports für die Budgetüberwachung.

## Schlechte Toolintegration

Viele User wünschen sich eine bessere Integration von VCR mit anderen Tools. Sie sind für das ausüben ihrer Rolle auf verschiedene Werkzeuge angewiesen, haben aber oft keine leicht zu bedienende Schnittstelle um ihre Daten aus VCR weiterzuverwenden. Dies geht so weit, dass viele User nicht nur die Excelexports nutzen, sondern auch Daten aus der Webansicht direkt in Excel kopieren, um mit den dort angezeigten Daten weiterarbeiten zu können. Dies betrifft vor allem das erstellen von aggregierten Projektdaten, weil dies in der Oberfläche von VCR nicht möglich ist. Aber auch andere Schnittstellen sind erwünscht, da zum Beispiel die Verwendung von Jazz Daten in Reports nur durch einen manuellen Export / Import möglich ist.

Mit einer besseren Anbindung an andere Tools könnten sich viele Nutzer Arbeit ersparen, welche sie heute doppelt verrichten müssen. So könnten sie sich auf wesentlichere Aspekte ihrer Aufgabe konzentrieren.

## Rechtmanagement

In VCR gibt es heute nur sehr limitiertes Benutzerrechtmanagement. Als Folge dieser Einschränkung gibt es heute über 100 Benutzer mit Administrationsrechten. Viele Vorgänge können nicht mit normalen Benutzerrechten durchgeführt werden, und erfordern dann direkt Administrationsrechte. Zum Beispiel können einige Datenimporte nur mit Administrationsrechten vorgenommen werden. Mit einer feingranularen Rechtevergabe könnte dieses Problem vermieden werden. Eine zukünftige Lösung sollte eine verfeinerte Rechtevergabe anbieten.

## Notwendigkeit für Excel

Durch die schlechte Integration mit anderen Tools und die limitierten Webansichten nutzen viele Leute Excel, um fehlende oder nicht genau passende Funktionalitäten nachzubauen. Weit verbreitet ist das verwenden von exportierten Daten um speziell angepasste Dashboards und andere Ansichten zu bauen. Insbesondere eine fehlende Aggregation von angezeigten Daten veranlasst Benutzer dazu, auf Excel für die Datenverarbeitung zurückzugreifen. Für viele Daten bietet VCR auch keine Eingabemasken, womit Excel zu einer Datensenke wird. Daten werden ausserhalb von VCR aufbereitet, bleiben dann aber auch dort. So müssen Auswertung immer wieder vorgenommen werden, anstatt dass diese verarbeiteten Daten persistiert werden können.

Auch die nur teilautomatisierten Meilensteinbewilligungs- und Out Of Bounds Prozesse sind an Excel (und Outlook) gebunden. So werden Nutzer gezwungen, Workflows manuell ausserhalb des Tools durchzuführen.

## Mühsamer Datenimport

VCR bietet Importschnittstellen für einige Daten, beispielsweise eine Schnittstelle um bearbeitete SAP Daten zu importieren. Daten in VCR zu importieren ist jedoch ein mühsamer Prozess, da die Schnittstellen nicht fehlertolerant sind, und in der Regel nur schlechte Fehlermeldungen bieten. So kann es zahlreiche Versuche benötigen, um herauszufinden wie eine Importdatei verändert werden muss um das System zufriedenzustellen.

Viele User sind jedoch auf diese Schnittstellen angewiesen, da es entweder die einzige Möglichkeit darstellt benötigte Daten im System bereitzustellen, oder aber die externe Aufbereitung der Daten deutlich bequemer ist. Durch verbesserte Eingabemasken in der Webansicht könnte vor allem die Datenaufbereitung im kleinen Rahmen direkt in VCR ermöglicht werden.

## Schlechte Performance

Zahlreiche Benutzer, vor allem diejenigen die in Ansichten mit vielen Projekten arbeiten, beklagen sich über Performanzprobleme. Dies äussert sich in sehr langen Ladezeiten bei diversen Vorgängen, zum Beispiel beim Laden der Hierarchien, oder durch lange dauernde Reportprozesse. Dies dürfte vor allem in der nachträglich eingebauten Datenhistorie begründet sein. Auch durch diese Probleme werden Benutzer dazu verleitet, das Tool weniger zu nutzen und möglicherweise ihre Arbeit in Excel zu erledigen.

## 9.3. Bestandsanalyse

*Dieses Kapitel ist ein Auszug aus [\[SA-23\]](#)*

Zu Beginn haben wir uns einen Überblick über die vorhandenen Systeme geschaffen. Nachfolgend sollen exemplarisch die wichtigsten Eckpunkte dieser Systeme vorgestellt werden, um ein besseres Verständnis für die Bestandssysteme zu erhalten.

Wir haben uns nachfolgend auf diejenigen Systeme beschränkt, welche wir mit unserem Prototyp primär tangieren.

### 9.3.1. VCR

Schon von Beginn weg war klar, dass VCR heute eine zentrale Rolle spielt. Aus diesem Grund sollen nachfolgend der Funktionsumfang im Allgemeinen und einige Funktionen im Speziellen etwas näher beleuchtet werden.

#### Funktionsumfang

VCR beinhaltet eine Vielzahl an Funktionen. Dies ist alleine schon an der dreistufigen Navigation (horizontales Menüband, vertikale Menüs und Tabs) ersichtlich.

Ob und in welchem Umfang all diese Funktionen heute nicht genutzt werden, konnte uns niemand abschliessend sagen. Aus diesem Grund wurde nun ein Auditing in VCR eingebaut, welches im Januar 2019 ausgerollt wird. In einigen Monaten lässt sich somit sagen, welche Teile der Applikation auch heute noch aktiv und intensiv genutzt werden.

#### Einstieg

Der Einstieg in die Applikation kann nur auf eine Art erfolgen, da die Applikation kein Routing unterstützt. Es ist also nicht möglich, über einen Direktlink auf eine beliebige Seite zu kommen. Jeder Nutzer landet zuerst auf der Projektübersicht. Diese wird nach seinem zuletzt als *Default* festgelegten Filter erstellt.

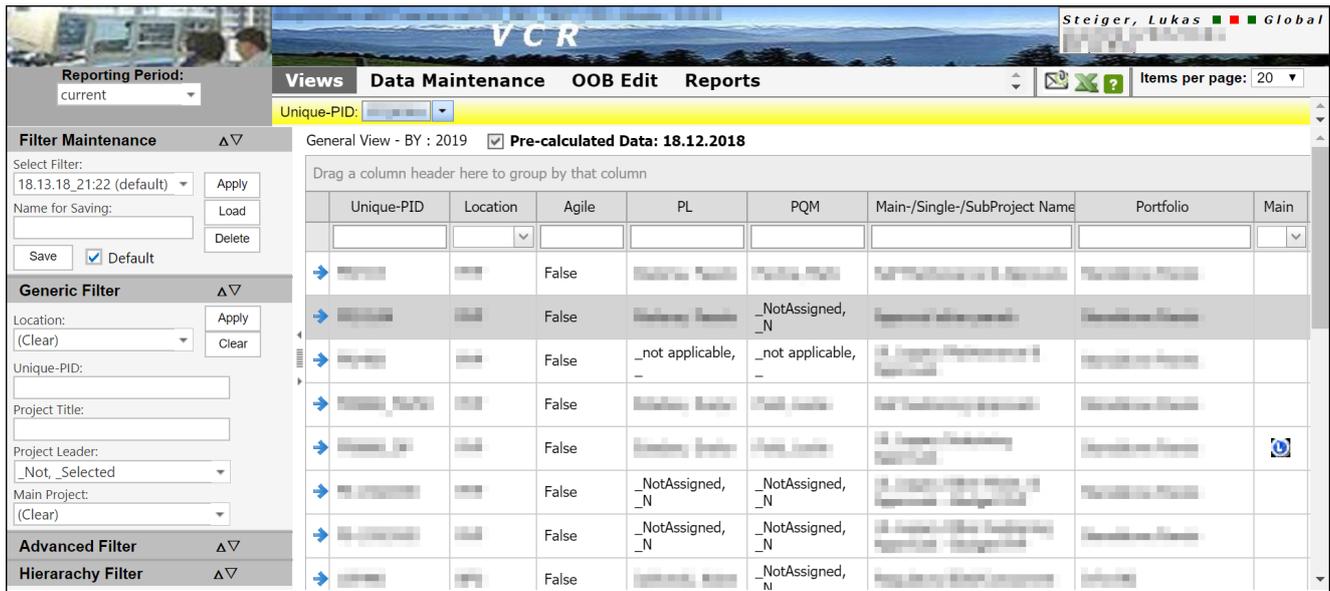


Abbildung 65. Einstiegsseite, der Standardfilter des Nutzers ist angewendet

### Projektliste

Ist die Projektliste einmal angezeigt, hat der Nutzer die Möglichkeit, durch Modifikation der zahlreichen Filter auf der linken Seite den Inhalt der Projektliste auf fast jede erdenkliche Art zu filtern. Die Liste selbst kann auch sortiert und gefiltert werden.

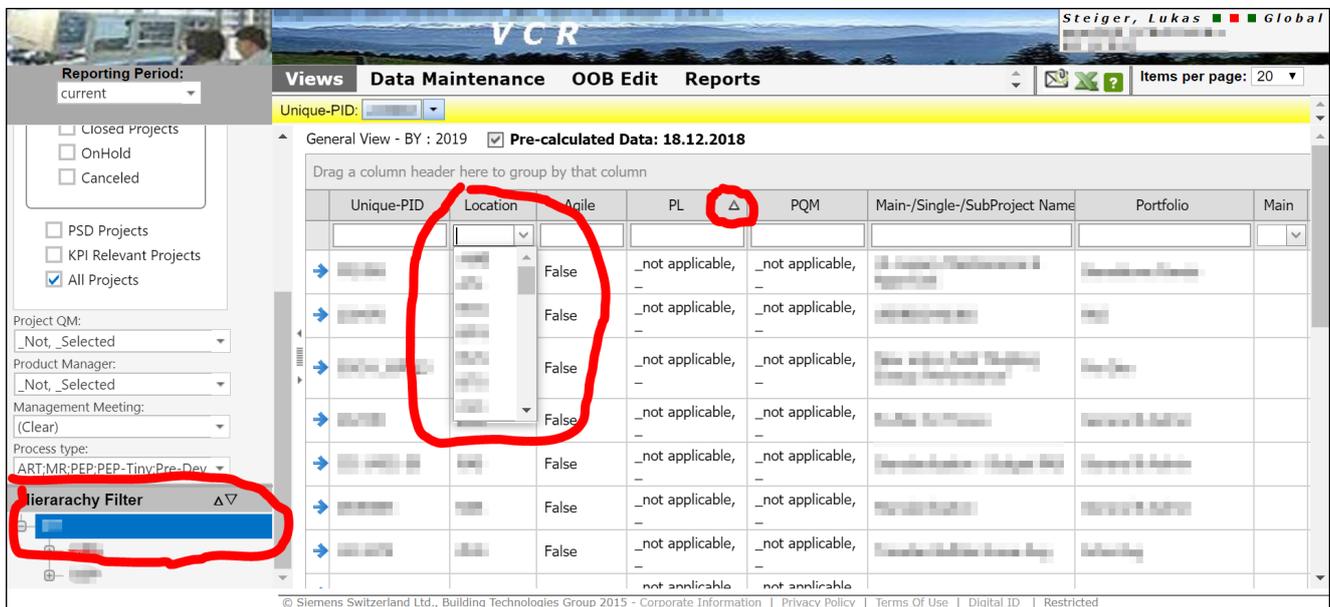


Abbildung 66. Die Projektlistenansicht lässt sich beliebig filtern

Durch einen Klick auf den blauen Pfeil in der Liste lässt sich die nachfolgend erläuterte Projektübersicht aufrufen.

## Projektübersicht & Verwaltung

In der Projektübersicht gibt es u.a. den Reiter *Identification*. Hier sind die wichtigsten Projektmetadaten abgelegt. Unter dem Reiter *Notes* können Projektnotizen abgelegt werden, auf dem Reiter *Overview* sind einige Textfelder, z.B. die Projektbeschreibung, zu finden.

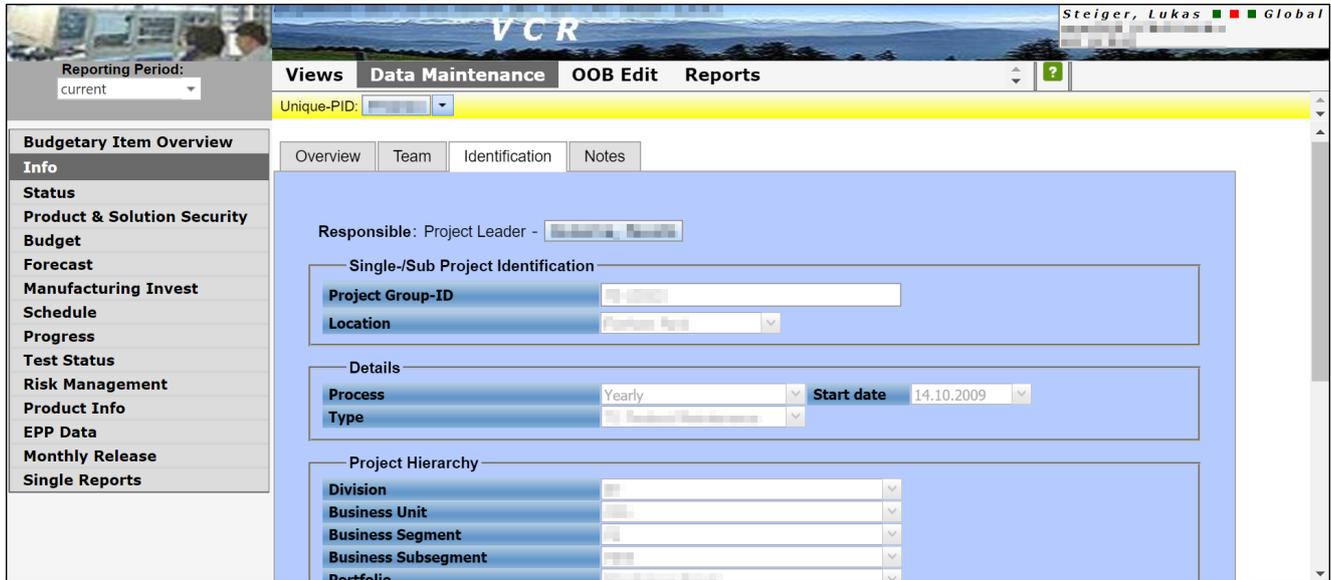


Abbildung 67. Informationen zur Projektidentifikation

Im gleichen Menü, jedoch im Reiter *Team*, sind die Teammitglieder und deren Rollen definiert. Die Rollenzuweisungen können hier auf der Zeile editiert werden.

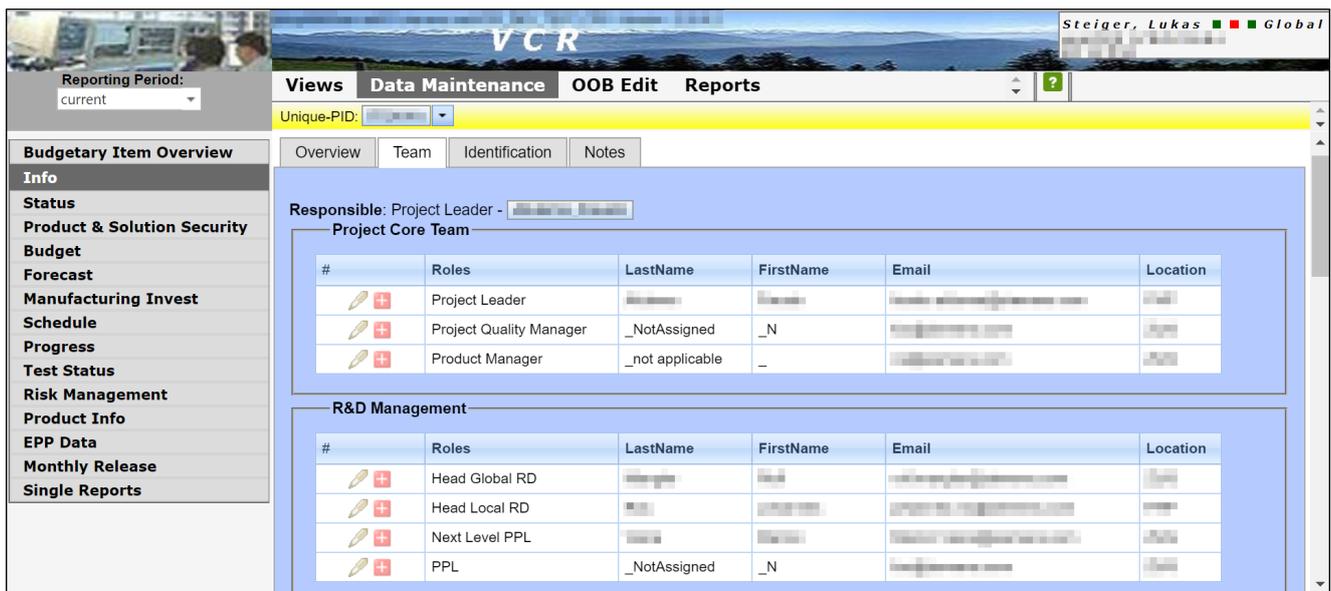


Abbildung 68. Gruppierte Rollenübersicht mit Möglichkeit zur Änderung der Zuordnung

In der Meilensteinübersicht (*Schedule*) sind die geplanten Meilensteine ersichtlich. Zudem ist ersichtlich, ob der aktuelle Planungsstand noch mit dem zuletzt bewilligten Stand übereinstimmt. Für die Bewilligung eines Meilensteins muss jedoch auf das *Meilenstein*

Approval Sheet zurückgegriffen werden (dieses wird auf den folgenden Seiten noch genauer durchleuchtet).

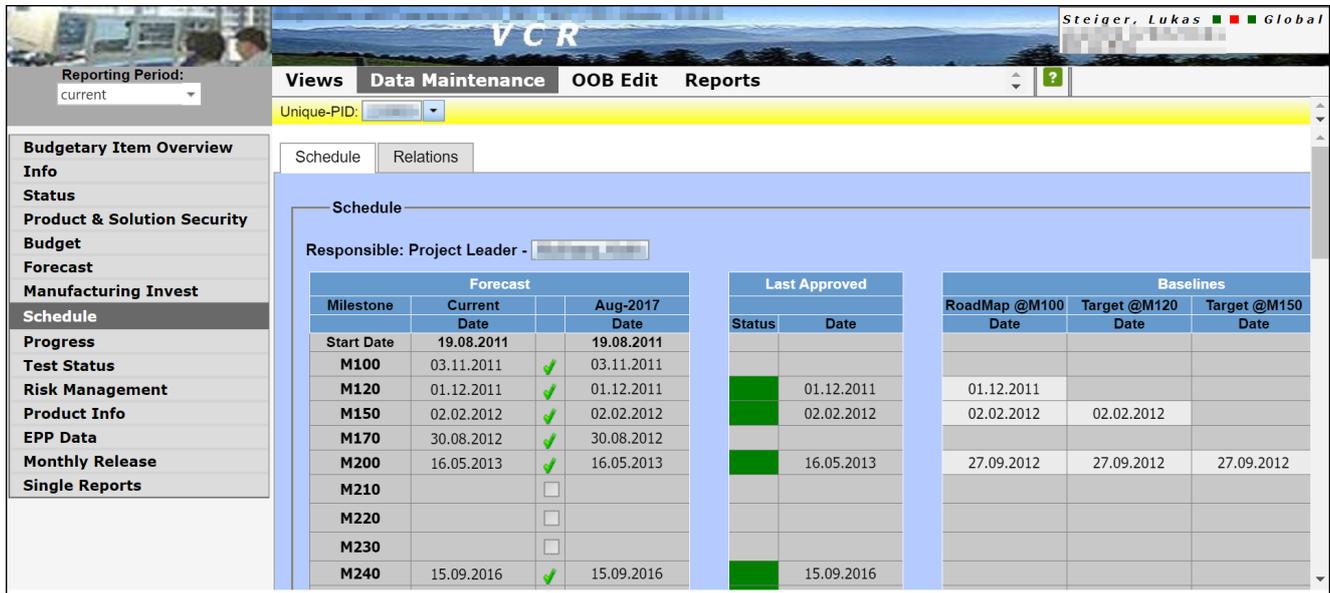


Abbildung 69. Meilensteinübersicht des Projektes

### Andere wichtige Funktionen

Folgende Funktionalität erachten wir ebenfalls als wichtig, sind hier jedoch nur stichwortartig erläutert:

#### Reports

Generierung von projektübergreifenden, vordefinierten Reports als PDF und/oder Excel. Diese Reports sind die einzige Möglichkeit, um projektübergreifende, aggregierte Informationen zu erhalten.

#### OOB Edit

Erstellen und Modifizieren von Change Requests für Budget und/oder Meilensteintermine. Für den Bewilligungsprozess muss jedoch auf das *Budget & Schedule Change Request (OOB) Sheet* zurückgegriffen werden.

#### Data Maintenance → Budget

Beherbergt das bewilligte Gesamtprojektbudget sowie die Aufteilung dieses auf die verschiedenen Geschäftsjahre.

#### Data Maintenance → Forecast

Plan-/Ist Vergleich des Projektbudgets

### **Data Maintenance → Risk Management**

Liste mit allen Projektrisiken und deren Bewertung sowie Massnahmen zur Minderung oder Eliminierung der jeweiligen Risiken.

### **Data Maintenance → Monthly Release**

Schnappschuss von bestimmten Daten auf monatlicher Basis. Kann entweder manuell ausgelöst werden oder wird automatisch gemacht in bestimmten Intervallen.

### **Data Maintenance → Single Reports**

PDF und Excel Reports für das aktuelle Projekt, u.a. Generierung des *Milestone Approval Sheet*.

## **9.3.2. Meilenstein Approval Sheet**

### *Hinweis*

Dieses Unterkapitel ist hier weggelassen, da dieser Ablauf in aller Tiefe im Kapitel [Meilensteinbewilligungsprozess](#) behandelt wird.



## 9.4. Personas

Dieses Kapitel ist ein Auszug aus [\[SA-24\]](#)

Auf Basis der geführten Interviews, den diversen Gesprächen mit unseren Auftraggebern und den Schilderungen einiger anderer Personen haben sich für uns die wichtigsten Benutzergruppen herauskristallisiert. Nachfolgend soll diesen Personen ein *Gesicht* in Form von Personas gegeben werden (Name, Alter und Geschlecht sind zufällig gewählt).

### 9.4.1. Quentin - Der Qualitätsbewusste

<b>Name</b>	Quentin
<b>Alter</b>	32
<b>Funktion &amp; Aufgaben</b>	Soll sicherstellen, das sämtliche Projekte den Qualitätsansprüchen der Firma genügen. Er begleitet Audits und erzeugt Projektmetriken über die Organisation hinweg.
<b>Funktion &amp; Aufgaben</b>	Hat ein breites Grundwissen in Statistik und Datenanalyse. Nutzt heute Excel intensiv, um sich seine eigenen Sichten auf die Projektdaten einer ganzen Organisationseinheit zu erstellen.
<b>Erwartungen</b>	Wünscht sich aggregierte Projektinformationen auf jeder Organisationsstufe und die Möglichkeit, eigene Sichten mit eigens ausgewählten Projekten zu haben. Möchte mehr Zeit für die Analyse von Problemen in konkreten Projekten haben anstatt diese Zeit mit der Sammlung und Aggregation von Daten zu verbringen.
<b>Vorgehen</b>	Bei Ungereimtheiten, welchen auf einer höheren Stufe sichtbar werden, macht er häufig ein Drill-Down, um dem Problem auf den Grund zu gehen. Schaut sich auch gerne Details zu Projekten während den <i>PSD Meetings</i> an.



[\[Pixabay-2\]](#)

## 9.4.2. Praveen - Der doppelte Projektleiter

<b>Name</b>	Praveen
<b>Alter</b>	54
<b>Funktion &amp; Aufgaben</b>	Projektleiter von zwei Projekten. Seine Teammitglieder sitzen in Indien, den USA und der Schweiz. Eines seiner Projekte ist die Weiterentwicklung eines strategisch wichtigen und sehr zukunftsweisenden Produkts. Es wird mit agilen Projektmethodiken geführt. Im zweiten Projekt betreut er noch den Vorgänger des neuen Produkts. Hier werden nur noch Fehlerbehebungen und kleine Erweiterungen gemacht. In diesem Projekt wird nach einem Wasserfallprozess gearbeitet.
<b>Funktion &amp; Aufgaben</b>	Ist sich die Bedienung verschiedenster Softwarewerkzeuge gewohnt, benötigt zum Verständnis des Tools jedoch Schulungsunterlagen und kontextbezogene Hilfe.
<b>Erwartungen</b>	Informationen über den Zustand seiner Projekte sollen an einem zentralen Ort in aggregierter Form verfügbar sein. Jedes Tool soll seinem Zweck entsprechend genutzt werden, eine Integration zwischen den Tools ist ihm ein grossen Anliegen, um Doppelspurigkeiten zu minimieren.
<b>Vorgehen</b>	Für die Fortschrittsverfolgung nutzt er RTC, seine personellen Ressourcen plant er mithilfe von RMT. Für alles was nicht direkt durch ein Tool angeboten wird nutzt er Excel.



[Pixabay-1]

### 9.4.3. Monera - Die Finanzchefin

<b>Name</b>	Monera
<b>Alter</b>	37
<b>Funktion &amp; Aufgaben</b>	<p>Als <i>Controllerin</i> ist sie das finanzielle Gewissen ihres Segments. Sie beschäftigt sich so gut wie nur mit den finanziellen Aspekten eines Projektes. Budgetplanung für das zukünftige Geschäftsjahr, Nachverfolgung der aktuellen Plan- und Ist Kosten gehören genau so zu ihren Aufgaben wie nationale Aspekte wie Versteuerung und Verrechnung.</p>
<b>Funktion &amp; Aufgaben</b>	<p>Hat fundierte Kenntnisse in BI, Statistik und im Finanzwesen. Hat jedoch auch heute noch Mühe mit den existierenden Workflows, da viele Daten von Hand transformiert, importiert, exportiert und bearbeitet werden müssen.</p>
<b>Erwartungen</b>	<p>Die vielen manuellen und fehleranfälligen Schritte bei der Verarbeitung von Finanzdaten aus SAP, VCR, C3 und BPRS soll besser koordiniert und wo möglich automatisiert werden. Mit welchem Programm ist sekundär, Hauptsache es ist weniger fehleranfällig und zeitaufwendig.</p>
<b>Vorgehen</b>	<p>Hat viele Vorgaben von den übergeordneten Finanzstellen, in welche Daten in welchem Format zur Verfügung gestellt werden müssen. Ihr Ziel ist immer, die gewünschten Daten schlussendlich im gewünschten Format zu haben. Aufgrund ihrer guten Excel Kenntnisse ist dies ihr präferiertes Tool.</p>



[Pixabay-3]

#### 9.4.4. Olivia - Die Portfolio Expertin

<b>Name</b>	Olivia
<b>Alter</b>	48
<b>Funktion &amp; Aufgaben</b>	Als Portfolioleiterin verantwortet sie eine ganze Menge von Projekten. Sie koordiniert die vielen Projekte auf einer höheren Stufe und bringt diese in Einklang mit der Geschäftsstrategie.
<b>Funktion &amp; Aufgaben</b>	Hat ein grosses Domänenwissen zu den entsprechenden Projekten und Produkten. Kennt sowohl die Wünsche der Kunden wie auch die Herausforderungen in den Projekten. Bringt aus ihrer früheren Rolle als Projektleiterin auch konkrete Projekterfahrung mit.
<b>Erwartungen</b>	Aggregierte Sichten auf ihr Portfolio sind essenziell für sie, um einen gesamtheitlichen Überblick auf ihr Portfolio zu haben. Für Budgetdiskussion, Fortschrittsnachverfolgung oder frühzeitige Problemerkennung sind solche Sichten wichtig.
<b>Vorgehen</b>	Oft kommen Projektleiter mit Herausforderungen und Wünschen auf sie zu, z.B. mit dem Wunsch nach mehr Budget. Oft muss sie innerhalb ihres Portfolios Budget und Ressourcen verschieben, um die Geschäftsziele optimal zu erreichen.



[Pixabay-4]

## 9.4.5. Colin - Der Projektcoach

<b>Name</b>	Colin
<b>Alter</b>	57
<b>Funktion &amp; Aufgaben</b>	<p>Als PQM (Projektqualitätsmanager) werden Personen wie er bezeichnet, welche als Projektcoach tätig sind. Er kennt die Geschäftsprozesse zum einen und die verfügbaren Tools zum anderen und unterstützt mit diesem Wissen die Projekte bei der Anwendung dieser.</p>
<b>Funktion &amp; Aufgaben</b>	<p>Als ehemaliger Projektleiter kennt er die Nöte und Herausforderungen im Alltag und kann mit seiner Erfahrung wertvolle Unterstützung bieten. Aufgrund der fundierten Kenntnisse von Prozess und Tools kennt er die Vor- und Nachteile wie kaum ein anderer.</p>
<b>Erwartungen</b>	<p>Da er viele Projekte betreut, wünscht er sich eine Übersicht über <i>seine</i> Projekte und deren Status. Auch offene Aufgaben (z.B. Meilenstein Approvals) möchte er an einem zentralen Ort sehen. Die Daten müssen einfach exportiert werden können, damit er spezielle eigene Analysen machen kann.</p>
<b>Vorgehen</b>	<p>Informiert sich regelmässig proaktiv über den Status seiner Projekte, wobei er dabei heute auf div. Tools zurückgreifen muss. Versucht ab und an über ein Mobilgerät die zur Verfügung stehenden Informationen abzugreifen. Aufgrund der schlechten Mobiltauglichkeit zahlreicher Datenquellen (VCR, Jazz, Excel) gibt er aber jedes mal wieder frustriert auf.</p>



[\[Pixabay-5\]](#)

## 9.5. Benutzungsszenarien

Dieses Kapitel ist ein Auszug aus [\[SA-25\]](#)

### 9.5.1. Szenario 1: 'Neues Projekt planen'

#### Ausführende Persona

Colin

#### Trigger & Ziel

Das Management hat in seiner letzten Sitzung entschieden, im neuen Geschäftsjahr ein neues, strategisch wichtiges Projekt zu starten. Das Projekt soll in einem Entwurfsmodus angelegt werden, sodass es fortan zu Planungszwecken (insbesondere Budgetplanung) bereits genutzt werden kann, ohne dass jedoch bereits alle Angaben wie Projektleitung bekannt sein müssen.

#### Resultat

Das Projekt ist mit Namen, Standort, vorläufigem Zuständigen, sowie allfällig anderen wichtigen Metadaten erfasst. Es hat einen Status (z.B. Draft, Proposed) welcher klar ausdrückt, dass das Projekt erst in der Planungsphase steckt, d.h. noch nicht aktiv ist.

### 9.5.2. Szenario 2: 'Zeitplan abrufen'

#### Ausführende Persona

Quentin

#### Trigger & Ziel

Soeben hat er in der Kaffeeecke erfahren, dass das Projekt *"Neue Fernwartungssoftware"* während seiner Ferienabwesenheit bereits wieder einen neuen Zeitplan erhalten soll. Er möchte sich direkt im Tool eine Übersicht über den aktuellen Zeitplan verschaffen und nachschauen, ob der Projektleiter bereits einen angepassten, jedoch noch nicht bewilligten Zeitplan erfasst hat.

#### Resultat

In der Übersicht sieht er, dass alle geplanten Daten immer noch den zuletzt bewilligten Daten entsprechen. Es muss sich also entweder um ein Gerücht handeln, oder die Daten sind nicht aktuell, er schreibt deshalb dem Projektleiter eine Kurznachricht.

### 9.5.3. Szenario 3: 'Aktuelle Übersicht über meine Projekte'

#### Ausführende Persona

Olivia

#### Trigger & Ziel

Im Moment gibt es in ihrem Segment viele Projekte, welche auf dem kritischen Pfad sind. Sie will sich deshalb regelmässig den aggregierten Status ihrer Projekte anschauen, um proaktiv reagieren zu können. Dies macht sie am liebsten am Montagmorgen auf dem Weg zur Arbeit, bevorzugterweise auf einem ihrer Mobilgeräte.

#### Resultat

In der Übersicht sieht sie die wichtigsten Kerndaten aggregiert über alle Projekte hinweg. Plan-/Ist Vergleich von Budget und Meilensteinen, Anteil an Projekten, welche als kritisch gelten oder offene Change Requests.

### 9.5.4. Szenario 4: 'Diskrepanz identifizieren'

#### Ausführende Persona

Olivia

#### Trigger & Ziel

Beim betrachten des aggregierten Budgets fällt eine ausserordentlich hohe Diskrepanz zwischen Soll und Ist Budget auf. Sie möchte nun das Projekt finden, welches diese grosse Diskrepanz verursacht.

#### Resultat

Dank der Drill-Down Funktion findet Sie schnell das Projekt. Es hat ein mehr als 10 Mal grösseres Ist Budget als noch vor einer Woche. Sie vermutet einen Tippfehler und kontaktiert den PL.

### 9.5.5. Szenario 5: 'Meilensteinbewilligung erteilen'

#### Ausführende Persona

Colin

#### Trigger & Ziel

Erhält Benachrichtigung über eine angefragte Meilensteinbewilligung. Pflichtbewusst möchte er gleich den Projektstatus anschauen und die Genehmigung erteilen, sofern alles in Ordnung ist. Dazu wird der Link in der Benachrichtigung angeklickt.

#### Resultat

Sieht sich die wichtigsten Projektstatusinformationen an. Er sieht, dass noch wesentliche Informationen fehlen. Anstelle von einem 'Approval' hinterlässt er einen Kommentar, was ihm noch fehlt und setzt den Status für sich auf 'Abgelehnt'.

### 9.5.6. Szenario 6: 'Mehr Budget und Zeit beantragen'

#### Ausführende Persona

Praveen

#### Trigger & Ziel

Aufgrund von verschärften Normen, welche per 1. Januar aktiv werden, benötigt er mehr Zeit und Budget, um die entsprechenden Anpassungen an seinem Produkt normgerecht umsetzen zu können.

#### Resultat

Er nutzt ein Lesezeichen in seinem Browser, um direkt auf seine Projektübersicht zu gelangen. Er erfasst in der Meilensteinübersicht die neuen Termine und erhöht das Gesamtprojektbudget. Danach löst er einen Change Request aus.

### 9.5.7. Szenario 7: 'Meine offenen Aufgaben'

#### Ausführende Persona

Colin

#### Trigger & Ziel

Hat sich soeben wieder daran erinnert, dass er vor wenigen Tagen eine Benachrichtigung erhalten hat, um ein *Approval* zu geben. Da er aber auf ersten Anhieb die Benachrichtigung nicht findet, will er einfach direkt in seinem Dashboard nachsehen, was er denn noch zu erledigen hat.

#### Resultat

In seinem persönlichen Dashboard sieht er zwei offene Aufgaben. Durch einen Klick auf einen der Einträge landet er direkt in der entsprechenden Ansicht im jeweiligen Projekt.

### 9.5.8. Szenario 8: 'Top-Down Budgetplanung'

#### Ausführende Persona

Monera

#### Trigger & Ziel

Soeben wurde von der Konzernleitung das Divisionsbudget für nächstes Geschäftsjahr festgelegt. In einem ersten Schritt muss Sie das Budget auf die Segmente verteilen, entsprechend den Einigungen mit den Segmentleitern.

#### Resultat

In der hierarchischen, tabellarischen Budgetübersicht gibt Sie das Jahresbudget auf Divisionsebene ein und verteilt es dann auf die Segmente.