

Master of Science FHNW in Virtual Design and Construction (VDC)

Integration eines digitalen Bauwerksmodells in die Bauindustrie unter Verwendung der Low-Code-Technologie



Thesis Verfasserin

Leah Laura Lüscher

Thesis Begleiterin

Prof. Nora Dainton

Studiengangleiterin MSc FHNW Virtual Design and Construction

nora.dainton@fhnw.ch

Thesis Experten

Dr. Eder Martínez Head of Lean Civil Engineering

Felix Stauch Head of BIM Civil Engineering

Praxispartnerin

Implenia Schweiz AG

Integration eines digitalen Bauwerksmodelles in der Bauindustrie unter Verwendung der Low-Code-Technologie

© 2025 by Leah Lüscher is licensed under [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)



Namensnennung - Nicht-kommerziell - Keine Bearbeitung 4.0 International Public License

Alle Rechte, insbesondere das Recht der Vervielfältigung, Verbreitung und Übersetzung werden vorbehalten. Kein Teil des Werkes darf in irgendeiner Form ohne schriftliche Genehmigung des Verfassers kommerziell reproduziert oder über elektronische Systeme für kommerzielle Zwecke verbreitet werden. Die Genehmigung ist beim Verfasser einzuholen.

Eigenständigkeitserklärung

Ich erkläre hiermit,

dass ich die vorliegende Master-Thesis mit dem Titel «**Integration eines digitalen Bauwerksmodells in die Bauindustrie unter Verwendung der Low-Code-Technologie**» selbst und selbständig verfasst habe,

dass ich sämtliche nicht von mir selbst stammenden Textstellen bzw. Bestandteile eines Werkes (Bilder, Grafiken, Codes, etc.) gemäss gängigen wissenschaftlichen Zitierregeln korrekt zitiert und die verwendeten Quellen gut sichtbar erwähnt habe;

dass ich in einem Verzeichnis alle verwendeten Hilfsmittel (KI-Assistenzsysteme wie Chatbots [z.B. ChatGPT], Übersetzungs- [z.B. DeepL] Paraphrasier- [z.B. Quillbot]) oder Programmierapplikationen [z.B. Github Copilot] deklariert und ihre Art der Verwendung offenlege und bei den entsprechenden Textstellen angegeben habe,

dass ich sämtliche immateriellen Rechte an von mir allfällig verwendeten Materialien wie Bilder oder Grafiken erworben habe oder dass diese Materialien von mir selbst erstellt wurden;

dass das Thema, die Arbeit oder Teile davon nicht bei einem Leistungsnachweis eines anderen Moduls verwendet wurden, sofern dies nicht ausdrücklich mit der Dozentin oder dem Dozenten im Voraus vereinbart wurde und in der Arbeit ausgewiesen wird;

dass ich mir bewusst bin, dass meine Arbeit auf Plagiate und auf Drittautorschaft menschlichen oder technischen Ursprungs (künstliche Intelligenz) überprüft werden kann;

dass ich mir bewusst bin, dass die Hochschule für Architektur, Bau und Geomatik einen Verstoss gegen diese Eigenständigkeitserklärung bzw. die ihr zugrundeliegenden Studierendenpflichten der Studien- und Prüfungsordnung der Hochschule für Architektur, Bau und Geomatik verfolgt und dass daraus disziplinarische (Verweis oder Ausschluss aus dem Studiengang) Folgen resultieren können.

Vorname Nachname:

Leah Lüscher

Ort, Datum:

Reinach AG, den 17. September 2025

Unterschrift:

Abstract

Die Bauindustrie befindet sich weitgehend in der digitalen Transformation. Obwohl Building Information Modeling die datenbasierte Abbildung eines Bauwerks während des gesamten Lebenszyklus ermöglicht, stossen die Nutzenden an ihre Umsetzungsgrenzen durch zu komplexe Alltagsanwendungen. In diesem Kontext gewinnen Low-Code-Plattformen an Aufmerksamkeit, da diese Technologie die Entwicklung anwendungsspezifischer Softwarelösungen auch ohne tiefgehende Programmierkenntnisse und mit einfach verständlichen Code-Bausteinen ermöglicht.

Mit dieser Arbeit wird untersucht, wie sich ein digitales Bauwerksmodell und Low-Code-Plattformen kombinieren lassen, um die Effizienz und Innovationskraft in der Bauwirtschaft zu steigern. Auf Basis einer Literaturrecherche und unstrukturierter Experteninterviews werden die Chancen und Herausforderungen der kombinierten Nutzung mittels Prototyps analysiert. Anschliessend erfolgt eine Diskussion und Nutzwertanalyse im Rahmen einer Fokusgruppe zur Validierung des Prototyps.

Experteninterviews zeigen, dass Low-Code bislang vor allem für interne Prozessoptimierungen eingesetzt wird, während Themen wie Data Governance und Skalierbarkeit brisant sind. In der Planung werden Dashboards zum Vergleich von Projektdaten oder Kosten erstellt, zudem erfreuen sich Apps zur Bewirtschaftung kleiner Datenbanken auf der Baustelle grosser Beliebtheit und werden vorzugsweise mit Microsoft Power Apps umgesetzt.

Als Produkt wird eine Applikation entwickelt, mit der der Baufortschritt bauteilgenau im Rahmen eines Baujournals mobil bearbeitbar ist. Mit diesem Werkzeug lassen sich Anpassungen in der Planung durch kürzere Abstände zwischen dem Abgleich von IST und SOLL agil steuern und umsetzen. Durch das standardisierte Datenschema IFC sind die Daten für alle Beteiligten zugänglich. Dadurch wird die Schnittstelle zwischen Baumanagement und Planung geschlossen, was für mehr Sicherheit und eine verbesserte Zusammenarbeit sorgt. Ebenso wird durch die Berücksichtigung der normativen Grundlagen eine skalierbare Applikation geschaffen.

Low-Code-Plattform, IFC, digitales Bauwerksmodell, Baufortschritt, Baujournal, Microsoft Power Apps

Danksagung

An dieser Stelle danke ich der Firma Implen AG, namentlich dem Thesis Experten Herr Dr. Eder Martínez für die fachliche Unterstützung bei meiner Erarbeitung. Der Firma Implen für die Fachexpertise in der Fokusgruppe. Zudem den Interview-Partnern, welche mir ehrliche Einblicke gaben und viele ihrer Applikationen gezeigt und vorgeführt haben.

Ein besonderer Dank geht an meinen Thesis Begleiter Prof. Nora Dainton, der Institutsleiterin vom Institut Digitales Bauen der Fachhochschule Nordwestschweiz, welche mich von Anfang an unterstützt hat und mit ihrem Wissen und andere Sichtweisen in die Thesis eingebracht hat.

Mein Dank gilt im Weiteren den Personen*, die mich durch ihre Unterstützung auf dem Weg begleitet haben. Den.

Meinem Partner Thierry-Benoît Wälchli verdanke ich die Unterstützung und freie Minuten, in welchen er sich um unsere Élodie gekümmert hat. Und welchen ich zu jeder Tages- und Nachtzeit um Rat und seine Expertisen bitten durfte. Und natürlich auch Élodie selbst, die ihre Mama zwar nicht geschont hat, sie aber trotzdem jeden Morgen mit einem Strahlen entzückt hat.

Inhalt

1	EINLEITUNG	1
1.1	AUSGANGSLAGE UND MOTIVATION	1
1.2	PROBLEMSTELLUNG	1
1.3	ZIEL DER ARBEIT	2
1.4	ERWARTETE RESULTATE	2
1.5	ABGRENZUNG	2
1.6	AUFBAU UND STRUKTUR DIESER ARBEIT	3
2	STAND DER PRAXIS, TECHNIK UND DER FORSCHUNG	4
2.1	TECHNISCHE UMSETZUNG MIT DER LOW-CODE-TECHNOLOGIE	5
2.1.1	<i>Arten von Low-Code-Plattformen</i>	6
2.1.2	<i>Low-Coding vs. No-Coding</i>	8
2.1.3	<i>IFC und Low-Code-Plattformen</i>	9
2.1.4	<i>Autodesk Platform Services (APS)</i>	9
2.1.5	<i>Microsoft Power Platform</i>	10
2.1.6	<i>Model Driven Engineering (MDE)</i>	12
2.2	DIE VERSCHIEDENEN AKTEURE / BETEILIGTEN	13
2.2.1	<i>Perspektive Unternehmen</i>	14
2.2.2	<i>Perspektive Entwickelnde</i>	14
2.2.3	<i>Perspektive Nutzende</i>	15
2.3	GESETZLICHE UND NORMATIVE RAHMENBEDINGUNGEN	15
2.3.1	<i>Baujournal</i>	15
2.3.2	<i>Baufortschritt</i>	15
2.3.3	<i>Räumliche Strukturierung des Projektes</i>	16
2.4	PROJEKTPLANUNG AUF DER BAUSTELLE	16
2.4.1	<i>Last Planner System (Aufgaben / Aktivität)</i>	17
2.4.2	<i>Produktions- und Ablauforganisation</i>	18
2.4.3	<i>Baufortschritt im digitalen Bauwerksmodell (4D-BIM)</i>	19
2.4.4	<i>Applikationen zur Dokumentation vom Baufortschritt</i>	20
2.5	IFC-DATENSHEMA	21
2.5.1	<i>Räumliche Strukturierung des Projektes nach dem IFC-Datenschema</i>	21
2.5.2	<i>Räumliche Objekte im IFC-Datenschema</i>	22
2.5.3	<i>Bauprozessplanung im IFC-Datenschema</i>	22
2.5.4	<i>Bauprogramm im IFC-Datenschema</i>	23
2.5.5	<i>Datenaufbereitung vom IFC-Datenschema hin zur Weiterverarbeitung mittels Low-Code</i>	24
2.6	ERWÄHNUNGEN VON EXEMPLARISCHEN USE-CASES IN DER LITERATUR	25
2.6.1	<i>Automatisierung vom Baustelleninventar mittels Low-Code</i>	25

2.6.2	<i>Reservierung von Elektro-Ladestationen mittels Low-Code-Anwendung</i>	26
2.6.3	<i>Bauteilprüfung vor Ort mit einer Low-Code-Anwendung</i>	26
2.7	PERSPEKTIVEN DURCH DIE ANWENDUNG VON LOW-CODE	26
2.7.1	<i>Chancen durch den Einsatz von Low-Code-Technologien</i>	28
2.7.2	<i>Herausforderungen durch die Low-Code-Technologieanwendung</i>	29
2.8	KONKLUSION VOM STAND DER PRAXIS, TECHNIK UND DER FORSCHUNG	30
3	BESCHREIBUNG DES FORSCHUNGSBEDARFS	32
4	METHODIK	33
4.1	FORSCHUNGSDESIGN / FORSCHUNGSMETHODIK	34
4.2	ANALYSE	34
4.3	SYNTHESE	35
4.4	EVALUATION	36
5	ERGEBNISSE	37
5.1	INTEGRATION EINES DIGITALEN BAUWERKSMODELLS AUF EINER LOW-CODE-PLATFORM	37
5.1.1	<i>Exemplarisches Use-Case Baufortschritt im Schweizer Kontext</i>	37
5.1.2	<i>Vorgehensprozess zur Integration eines IFC in einer Low-Code-Plattform</i>	38
5.1.3	<i>Umsetzung Prototyp «Integration eines IFC in einer Microsoft Power App»</i>	39
5.1.3.1	Anforderungsdefinition	39
5.1.3.2	Fachlich-konzeptuelle Modellierung	42
5.1.3.3	Technische Implementierung	46
5.1.4	<i>Diskussionsergebnisse der Fokusgruppe in Bezug auf den Prototypen</i>	50
5.2	KONKRETE ANWENDUNGSSZENARIEN	54
5.2.1	<i>Applikationen</i>	55
5.2.2	<i>Dashboards und Automatisierungen</i>	56
5.2.3	<i>Technische Umsetzung</i>	58
5.3	CHANCEN UND HERAUSFORDERUNGEN AUS BAU-EXPERTENSICHT IN DER SCHWEIZ	59
5.4	ALTERNATIVE LOW-CODE-PLATFORM	61
6	DISKUSSION	63
6.1	TECHNISCHE VORAUSSETZUNGEN MIT IFC	63
6.2	INTEGRATION EINER IFC-DATEI IM ALLTAG	64
6.3	CHANCEN UND HERAUSFORDERUNGEN	64
6.4	KONKRETE ANWENDUNGSSZENARIEN	65
6.5	MICROSOFT POWER PLATFORM VS. AUTODESK PLATFORM SERVICES	66
7	FAZIT	67
7.1	SCHLUSSFOLGERUNG	68
7.2	AUSBLICK	68
8	QUELLENVERZEICHNIS	70
9	ABBILDUNGS- UND TABELLENVERZEICHNIS	77

9.1	ABBILDUNGEN	77
9.2	TABELLEN.....	78
10	ANHANG.....	79
	<i>Anhang B</i> Verzeichnis verwendete Hilfsmittel.....	80
	<i>Anhang C</i> Vorlage Baujournal.....	82
	<i>Anhang D</i> Extraktion von Daten aus dem IFC in ein CSV (Python-Code).....	83
	<i>Anhang E</i> Speicherung der Daten von der Datenbank in die IFC-Datei (Python-Code).....	87
	<i>Anhang F</i> Abstract English.....	90
	<i>Anhang G</i> Interview Totalunternehmer Schweiz.....	91
	<i>Anhang H</i> Interview beim Gleisbau-Unternehmen, Schweiz	92
	<i>Anhang I</i> Interview mit Totalunternehmer, Schweiz.....	93
	<i>Anhang J</i> Interview mit Totalunternehmer, Schweiz.....	95
	<i>Anhang K</i> Interview bei Bauleitung in einer Totalunternehmung, Schweiz.....	95
	<i>Anhang L</i> Interview beim Totalunternehmer, Schweiz.....	96
	<i>Anhang M</i> Interview beim Elektrotechnik-Unternehmen, Schweiz.....	96
	<i>Anhang N</i> Interview mit Digitalen Spezialisten für das Bauwesen, Dänemark.....	97
	<i>Anhang O</i> Interview mit einem Digital Consulting-Unternehmen, Schweiz	108
	<i>Anhang P</i> Link Prototyp.....	110

1 Einleitung

1.1 Ausgangslage und Motivation

Die Bauindustrie erlebt derzeit einen Wandel, der durch die Digitalisierung geprägt ist. Traditionelle Planungs- und Ausführungsprozesse stossen zunehmend an ihre Grenzen, wenn es um Effizienz, Transparenz und die Zusammenarbeit zwischen den Projektbeteiligten geht. Die Nutzung digitaler Bauwerksmodelle bei Bauabwicklung mittels Building Information Modeling (BIM) ermöglicht es, den gesamten Lebenszyklus eines Bauwerks von der Planung über den Bau bis zum Betrieb einheitlich und datenbasiert darzustellen. Gleichzeitig gewinnen Plattformen mit der Low-Code-Technologie als innovative Entwicklungsansätze an Bedeutung, da sie die Erstellung anwendungsbezogener Softwarelösungen ohne tiefgreifende Programmierkenntnisse ermöglichen. Die Kombination dieser beiden Technologien eröffnet neue Perspektiven für die Digitalisierung und Prozessoptimierung in der Bauwirtschaft.

Die Arbeit mit Low-Code ist laut Wong und Iijima in einigen Branchen bereits Teil des Arbeitsalltags und wird gemäss ihrer Prognose in Zukunft an Wichtigkeit gewinnen (Wong & Iijima, 2021). Dabei wird in den Bereichen Mobilfunk, Informations- und Telekommunikation sowie Gesundheit das höchste Wachstum erwartet. Laut einer Studie von Fortune Business Insights aus dem Jahr 2024 wird die Low-Code-Technologie im Baugewerbe allerdings nicht wirklich eingesetzt (fortune business insights, 2024). Auch die Verknüpfung mit der BIM-Arbeitsmethode bleibt bisher aus.

Digitale Bauwerksmodelle bietet in der Theorie zahlreiche Vorteile, doch ihre praktische Integration in die bestehenden Abläufe der Bauindustrie ist noch mit vielen Herausforderungen verbunden. Komplexe Informationstechnik (IT)-Systeme, mangelnde Schnittstellenkompatibilität und hohe Entwicklungskosten hemmen eine flächendeckende Implementierung. Gleichzeitig fehlt es vielen Unternehmen an personellen und technischen Ressourcen, um eigene, auf die spezifischen Anforderungen von Bauprojekten zugeschnittene Softwarelösungen zu entwickeln. Hier setzt die Low-Code-Technologie an: Sie ermöglicht es auch Akteuren ohne IT-Kenntnisse, digitale Anwendungen zur Verwaltung, Auswertung und Visualisierung von Bauwerksdaten zu erstellen. Die Integration digitaler Bauwerksmodelle über Low-Code-Plattformen könnte somit einen entscheidenden Beitrag zur Effizienzsteigerung, Qualitätsverbesserung und Innovationsförderung in der Bauindustrie leisten.

1.2 Problemstellung

Trotz des technologischen Fortschritts und des wachsenden Bewusstseins für die Vorteile digitaler Bauwerksmodelle hinkt die praktische Umsetzung in der Bauindustrie den Erwartungen hinterher. Ein zentrales Problem ist, dass bestehende Softwarelösungen zur Verwaltung und Integration von Bauwerksdaten meist **zu komplex, zu kostenintensiv** und nur **begrenzt anpassbar** auf die entsprechenden Geschäftsprozesse sind. Besonders kleinere und mittelgrosse Unternehmen (KMU) verfügen nicht über die personellen oder finanziellen Ressourcen, um massgeschneiderte IT-Systeme zu entwickeln oder diese umfassend in ihre Prozesse zu integrieren (Oberholzer, 2025).

Hinzu kommen Schwierigkeiten bei der Systemintegration und im Datenaustausch. Bestehende Anwendungen und Plattformen arbeiten häufig mit proprietären Datenformaten oder geschlossenen Systemarchitekturen. Das erschwert den reibungslosen Austausch zwischen den Projektbeteiligten. Die fehlende Interoperabilität führt zu Medienbrüchen, Doppelarbeiten und ineffizienten Arbeitsabläufen. Dadurch wird das Potenzial digitaler Bauwerksmodelle zur Prozessoptimierung und Qualitätssteigerung nicht ausgeschöpft (Spring, 2022)

Low-Code-Plattformen versprechen eine Lösung, da sie die Entwicklung individueller Softwareanwendungen ohne tiefgehendes Programmierverständnis ermöglichen. Doch auch hierbei stellen sich technische Fragen: Wie können Low-Code-Anwendungen sicher und stabil in bestehende IT-Landschaften eingebunden werden? Welche Schnittstellen sind erforderlich, um den Zugriff auf Bauwerksdaten zu gewährleisten? Oder wie kann die Bedienung so gestaltet werden, dass auch nicht-technisches Personal produktiv damit arbeiten kann?

Solche offenen Fragen verdeutlichen den Bedarf an praxisorientierten Konzepten, die sowohl die technischen als auch die personellen Voraussetzungen für eine erfolgreiche Integration digitaler Bauwerksmodelle mithilfe von Low-Code-Technologie berücksichtigen.

Weiter ist das Baugewerbe in Bezug auf Innovationen eher konservativ (outsystems, 2019). Die Gegenwart zeigt, dass unternehmensspezifische Geschäftsprozesse zwar optimiert wurden, diese jedoch im Sinne der digitalen Transformation zukünftig noch weiter vereinfacht werden müssen um den Herausforderungen bei der Ausführungseffizienz gerecht zu werden (Martinez, 2024a).

1.3 Ziel der Arbeit

Das Ziel dieser Master-Thesis ist die **Entwicklung einer Methode, mit der das Industry Foundation Classes (IFC)-Datenschema automatisch in die Microsoft Power Platform integriert werden kann**. Durch diese Integration soll es möglich werden, bauteilspezifische Informationen strukturiert und effizient aus einem digitalen Bauwerksmodell abzurufen und in einer anwenderfreundlichen Oberfläche bereitzustellen. Im Fokus steht die Frage, wie die technischen Strukturen und semantischen Inhalte des IFC-Standards in das Datenmodell der Power-Platform überführt werden können, um eine praxisnahe Anwendung im Baukontext zu ermöglichen.

Weiter soll untersucht werden, unter welchen technischen und organisatorischen Voraussetzungen eine solche Integration im operativen Arbeitsalltag der Bauindustrie sinnvoll einsetzbar ist. Ein besonderes Augenmerk liegt dabei auf der Identifikation von Chancen und Herausforderungen, die sich aus der Kombination von Low-Code-Technologie und offenen Bauwerksdatenformaten wie IFC ergeben.

1.4 Erwartete Resultate

Im Rahmen dieser Masterarbeit wird ein fundiertes und **praxisnahes Verfahren** geprüft, **mit dem sich ein IFC-Datenschema automatisiert in die Microsoft Power Platform**, spezifisch in Microsoft Power Apps, **integrieren lässt**. Mithilfe der prototypischen Umsetzung dieses Verfahrens soll demonstriert werden, wie bauteilspezifische Informationen aus einem digitalen Bauwerksmodell extrahiert, transformiert, strukturiert und innerhalb der Power Platform für Anwender zugänglich gemacht werden können.

Als weiteres Ergebnis wird die Entwicklung beziehungsweise der **Aufbau einer konkreten Power App** angestrebt. Diese soll demonstrieren, wie Bauwerksdaten nutzerfreundlich visualisiert und in den Arbeitsalltag eingebunden werden können. Die Anwendung soll es ermöglichen, relevante Bauteilinformationen gezielt abzufragen, darzustellen, sowie zu bearbeiten. Zudem sollen diese Daten vom Büroschreibtisch aus jederzeit abrufbar sein.

Im Verlauf der Arbeit sollen weiter **technische Herausforderungen und Schnittstellenprobleme identifiziert werden**, die bei der Integration der Systeme auftreten. Dies im Hinblick auf Datenstruktur, Application Programming Interface (API)-Nutzung oder Kompatibilität mit bestehenden IT-Lösungen. Mithilfe dieser Analyse sollen typische Stolpersteine benannt und Lösungsansätze skizziert werden, die eine erfolgreiche Umsetzung unterstützen können.

Ergänzend dazu erfolgt eine **Untersuchung**, unter welchen **technischen Rahmenbedingungen** eine derartige Lösung in der Praxis sinnvoll sowie langfristig tragfähig eingesetzt werden kann. Dazu gehören unter anderem Anforderungen an die interne IT-Infrastruktur, der Schulungsbedarf der Mitarbeitenden sowie notwendige Rollenverteilungen im Projektalltag.

Am Ende **werden die entwickelten Konzepte und technischen Umsetzungen mit den im Vorfeld ermittelten Praxisanforderungen abgeglichen**. Diese Gegenüberstellung erfolgt in enger Zusammenarbeit mit der Praxispartnerin sowie mit Fachexperten. Dadurch wird sichergestellt, dass die Ergebnisse nicht nur theoretisch tragfähig, sondern auch für reale Anwendungsfälle in der Bauindustrie relevant und nutzbar sind.

1.5 Abgrenzung

In dieser Arbeit liegt der Fokus auf dem Baufortschritt, der auf dem IFC-Datenschema basiert. **Andere Anwendungsbereiche oder IFC-Datenstrukturen**, die nicht unmittelbar damit in Zusammenhang stehen, werden nicht betrachtet.

Faktoren der Benutzerfreundlichkeit, rein visuelle No-Code-Lösungen und KI-gestützte Entwicklungsmethoden sind nicht Teil dieser Arbeit. Der Einsatz von künstlicher Intelligenz (z.B. maschinelles Lernen, Prognosemodelle) ist ebenfalls ausgeschlossen.

Die Untersuchung beschränkt sich auf die **Realisierungsphase** eines Bauprojektes. Frühere Phasen wie die Planung oder die Entwurfsentwicklung sowie spätere Phasen wie der Betrieb oder der Rückbau fallen nicht in den

Untersuchungsrahmen. Ebenso wird ausschliesslich des Hochbaus betrachtet, während Anwendungsfälle im Tiefbau (z.B. Strassen- oder Infrastrukturbau) nicht berücksichtigt werden.

Fragen des Datenmanagements, konkreter die Behandlung neuer IFC-Dateien beim Hochladen sowie die Sicherstellung der Datenaktualität innerhalb einer Applikation, sind nicht Gegenstand dieser Arbeit. Ebenso wird eine Rückführung der Daten in die jeweilige Autoren-Software (z.B. Revit, ArchiCAD) nicht untersucht.

Andere am Markt verfügbare Low-Code- oder No-Code-Plattformen werden ausschliesslich anhand von Literatur- und Quellenangaben beschrieben. Eine praktische Erprobung, Implementierung oder vergleichende Bewertung dieser Systeme erfolgt nicht.

1.6 Aufbau und Struktur dieser Arbeit

Die Abbildung 1 zeigt den schematischen Aufbau und die Struktur dieser Arbeit.

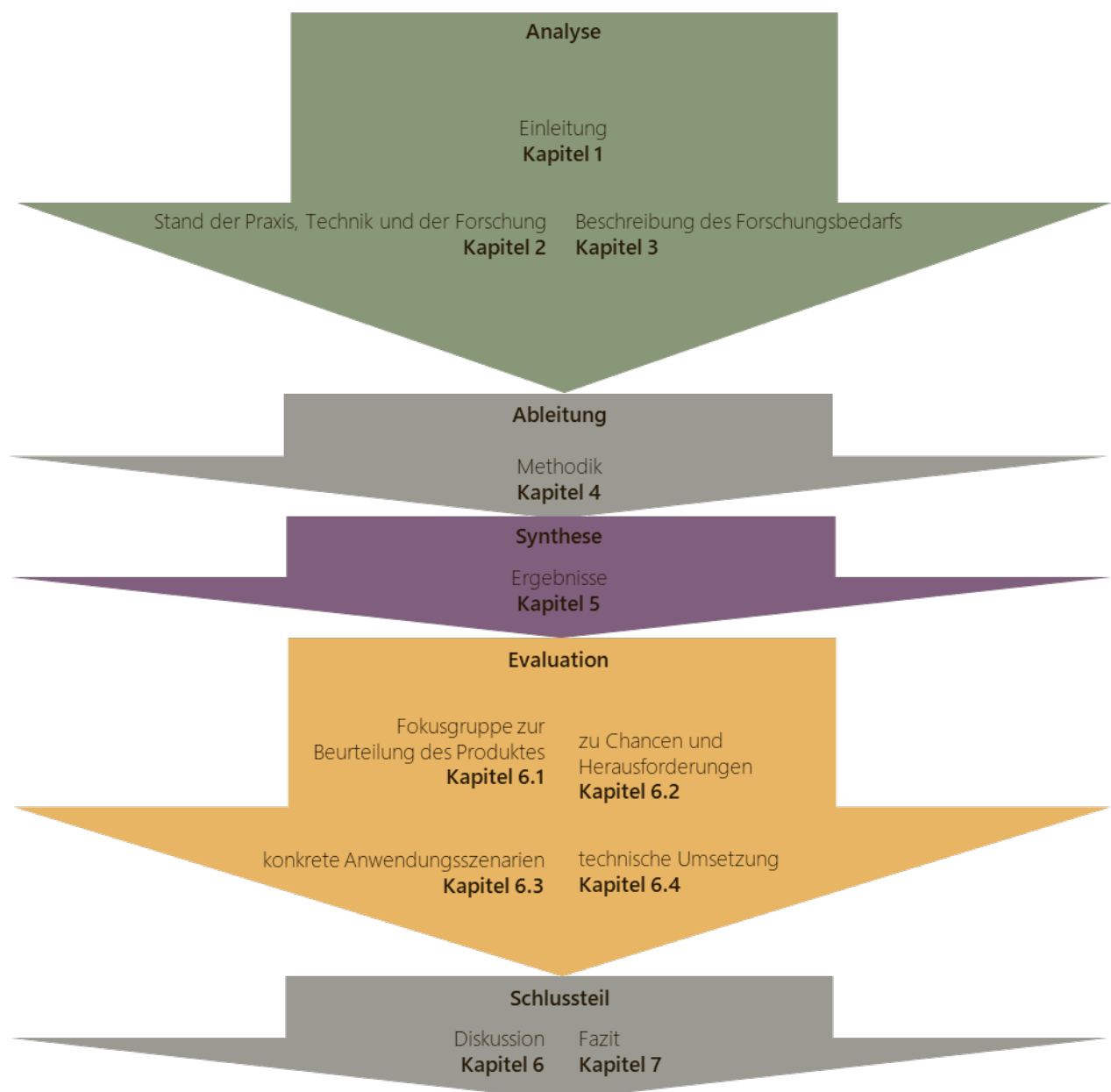


Abbildung 1, schematische Struktur der Master-Thesis (eigene Abbildung)

2 Stand der Praxis, Technik und der Forschung

In der Bau- und Immobilienentwicklungsbranche beispielsweise sind die Beziehungen zwischen den Akteuren durch kurzfristig am Markt orientierte Interaktionen geprägt und sind ein wichtiger Bestandteil des produktiven Sekundärsektors unserer Volkswirtschaft (Gann, 1996). Darin wird eine Politik des niedrigsten Preises gepflegt, bei der der Fokus auf Aktionen und kurzfristigen Gewinnen im operativen Geschäft liegt (Jacobsson & Linderoth, 2010). Dies führt dazu, dass die Akteure ihre eigenen Prozesse optimieren, anstatt den Bauprozess als Ganzes zu betrachten (Love et al., 1998). Die Optimierung der eigenen Prozesse der Akteure kann als Folge eines weiteren zentralen Merkmals betrachtet werden, durch das sich die Bau- und Konstruktionsbranche von anderen Branchen unterscheidet (Harty, 2005). Ein Bauprojekt wird zwar oft von einem Hauptauftragnehmer koordiniert, doch jede am Projekt beteiligten Organisationen hat ihren eigenen Einfluss darauf und bringt ihre eigenen Erwartungen und Arbeitsmethoden ein. Auch wenn vertragliche Verpflichtungen die Hauptstütze der formellen Koordination in der Branche sind, lassen sie sich in der Praxis nur schwer durchsetzen, besonders, wenn die Arbeiten bereits im Gange sind. (Linderoth et al., 2018, p. 3)



Abbildung 2, Softwarelandschaft und ihre Zuordnung (eigene Abbildung)

Die Abbildung 2 zeigt einen eigens erstellten Überblick über die Softwarelandschaft und deren Zuordnung zu den jeweiligen Technologiebereichen. Die Auflistung der Anwendungen ist nicht abschliessend und die Abgrenzungen sind meistens eher schwammig, da je nach genutztem Bereich andere Anwendungsstrategien genutzt

werden können. Die in **Fett** geschriebenen Anwendungen werden in dieser Thesis zu einem späteren Zeitpunkt umfassender beschrieben.

Der Begriff «Low-Code» (LC) ist kein neues Konzept, sondern ein Ansatz, der seit mehr als einem Jahrzehnt in der Praxis angewendet wird und im Jahr 2014 durch Forrester Research erstmals systematisch definiert wurde (Richardson & Rymer, 2014). Low-Code beschreibt eine Gruppe von Entwicklungswerkzeugen, mit denen sich Anwendungen über eine Drag-and-Drop-Oberfläche visuell erstellen lassen, ohne dass umfangreiche Programmierkenntnisse vorausgesetzt werden. Anstelle zeilenweise komplexen Codes in textbasierten Sprachen wie: Java, C oder Python zu schreiben, können Anwendungen mithilfe visueller Bausteine und vorkonfigurierter Module entwickelt werden (Hirzel, 2023; Revell, 2019, p. 1)

Low-Code-Entwicklungsplattformen (LCAP) sind darauf ausgerichtet, die manuelle Codierung zu minimieren und die Softwareentwicklung durch visuelle Schnittstellen zu vereinfachen. Die visuelle Programmierung ermöglicht es auch fachfremden Anwender:innen, funktionale Geschäftsanwendungen zu erstellen. Dies ist ein wichtiger Aspekt angesichts des wachsenden Bedarfs an individuellen Softwarelösungen bei gleichzeitig knappen IT-Ressourcen. Somit können Benutzer:innen mithilfe von Drag-and-Drop-Elementen nicht nur funktionale Steuerelemente wie Eingabefelder oder Navigationsschaltflächen in die Benutzeroberfläche integrieren, sondern lassen sich auch mittels vordefinierte Funktionsmodule zu vollständigen Anwendungen verbinden (Adrian et al., 2020, p. 46).

Diese Plattformen bieten Unternehmen eine ökonomischere und agilere Möglichkeit, auf die steigenden Anforderungen an digitale Lösungen zu reagieren (Sanchis et al., 2019, p. 2). Diese Entwicklung steht im Kontext einer zunehmenden Kluft zwischen dem steigenden Bedarf an digitalen Lösungen und der begrenzten Verfügbarkeit qualifizierter IT-Fachkräfte in den Unternehmen und in den Bildungssystemen (Metrolho et al., 2020). Der Fachkräftemangel stellt viele Unternehmen, besonders im Hinblick auf die Umsetzung digitaler Transformationsprojekte, vor grosse Herausforderungen (Bock & Frank, 2021, p. 734). Gleichzeitig steigt die Nachfrage nach individuell anpassbaren Anwendungen in Form mobiler Apps, die betriebliche Prozesse optimieren und die Effizienz steigern können. Weiter ermöglichen Low-Code-Development-Plattformen (LCDP) besonders in Branchen mit geringer IT-Kompetenz, wie dem Bauwesen, softwarebasierte Prozesse zu etablieren (Siddiqui et al., 2023). Unter diesen Aspekten gewinnen Low-Code-Development-Plattformen zunehmend an Bedeutung, da sie IT-Abteilungen entlasten und die Entwicklung digitaler Lösungen beschleunigen können (Bitkom, 2018; Mammadova et al., 2018).

In der Literatur wird jedoch darauf hingewiesen, dass die von Low-Code-Development-Plattformen erzeugten Codes häufig schwer nachvollziehbar sind, da sie oft ohne Dokumentation oder mit wenig aussagekräftigen Variablennamen erstellt werden (Woo, 2020).

Um diesen Entwicklungen adäquat begegnen zu können, rückt die Rolle innovativer Technologien wie Low-Code-Plattformen (LCP) zunehmend in den Fokus. Ihre wachsende Relevanz kann jedoch nicht isoliert betrachtet werden, sondern muss im grösseren Zusammenhang der digitalen Transformation verstanden werden. Die Untersuchung der Low-Code-Technologie erfolgt daher im Kontext dieses umfassenden Wandels. Osmundsen et al. definieren diesen Wandel als tiefgreifenden Veränderungsprozess, der durch digitale Technologien und Innovationen substantielle Auswirkungen auf Organisationen und ganze Branchen entfaltet (Osmundsen et al., 2018, p. 10).

2.1 Technische Umsetzung mit der Low-Code-Technologie

Low-Code-Plattformen (LCP) sind benutzerfreundliche technologische Innovationen, die verschiedene bekannte Systemdesignkomponenten in einer gemeinsamen Umgebung vereinen. Ein Hauptziel ist es, den Aufwand für Routineaufgaben bei der Entwicklung von Geschäftsanwendungen unter bestimmten, teilweise eingeschränkten Rahmenbedingungen zu verringern. Befinden sich die Anforderungen eines Projekts vollständig innerhalb des festen Rahmens einer LCP und sind die technischen sowie wirtschaftlichen Voraussetzungen erfüllt, können diese Plattformen die Produktivität der Softwareentwicklung deutlich erhöhen. (Bock & Frank, 2021, p. 739)

Eine zentrale technische Eigenschaft besteht in der Einbindung von Komponenten zur konzeptuellen Modellierung mit Notationen wie beispielsweise UML oder BPMN. (UML = Unified Modeling Language; BPMN = Business Process Model and Notation) Diese Werkzeuge verringern den Bedarf an klassischer Programmierung, da abstrakte Modelle direkt in ausführbare Anwendungen überführt werden können (Bock & Frank, 2021, p. 740). Technisch lassen sich dabei Überschneidungen mit anderen forschungsorientierten Ansätzen erkennen, auch wenn

deren Verfahren nur selten vollständig übernommen werden. Ähnlichkeiten bestehen hauptsächlich zu Konzepten der visuellen Programmierung (Costagliola et al., 2004; Ingalls et al., 1988).

Bemerkenswert ist hingegen, dass etablierte Forschungsansätze, wie etwa Referenzmodelle oder domänenspezifische Modellierungssprachen, bisher nur wenig Beachtung finden (Becker & Delfmann, 2007). Gerade diese Technologien könnten die Effizienz im Systemdesign erhöhen, da sie Domänenwissen direkt integrieren (Frank, 2013; Kelly & Tolvanen, 2008).

Technisch gesehen bieten LCP in der Regel eine Bibliothek mit grundlegenden Standardfunktionen, darunter vor allem mathematische Operationen, sowie vielfältige Möglichkeiten zur Anbindung externer Dienste über API. Die Integration erfolgt häufig über standardisierte Schnittstellen mittels Webservices oder Representational State Transfer (RESTful) Services. Viele Systeme unterstützen zusätzlich eine breite Auswahl an anbieterspezifischen APIs, beispielsweise Google API oder Social-Media-Schnittstellen. Weiter verfügen fast alle Plattformen über erweiterte Bereitstellungsfunktionen, deren Umfang und konkrete Ausgestaltung jedoch sehr unterschiedlich ausfallen können. (Bock & Frank, 2021, p. 740)

Ein wesentliches technisches Risiko bleibt jedoch der potenzielle Lock-in-Effekt: Modelle oder Anwendungen, die in einer LCP entwickelt wurden, können in den meisten Fällen nicht einfach auf eine andere Plattform übertragen werden. Dies beeinträchtigt den Investitionsschutz und kann zu technologischen Sackgassen führen. Ein exemplarisches Beispiel dafür ist die Einstellung von Googles AppMaker im Jahre 2021. (Bock & Frank, 2021, p. 738)

2.1.1 Arten von Low-Code-Plattformen

Obwohl viele LCP bestimmte technische Merkmale gemeinsam haben, gibt es erhebliche Unterschiede in Bezug auf deren: Funktionsumfang, Hauptzweck, Spektrum der beteiligten Technologien, Breite der Anwendbarkeit, Mittel der Gestaltung, Spezifikation und andere Aspekte. So lassen sich LCP gemäss Bock & Frank in vier grobe Kategorien einstufen (Bock & Frank, 2021, p. 735):

1. einfache Datenverwaltungsplattformen
2. klassische Workflow-Management-Systeme
3. erweiterte, grafische Benutzeroberflächen und datenzentrierte, integrierte Entwicklungsumgebungen (IDE, en, Integrated Development Environment)
4. komplexe Mehrzweckplattformen für die Konfiguration, Integration und Entwicklung von Geschäftsanwendungen.



Diese Einteilung bietet eine funktionale Perspektive auf die Vielfalt der existierenden Low-Code-Plattformen. Zusätzlich ermöglichen Marktanalysen eine Bewertung der Anbieter hinsichtlich ihrer strategischen Ausrichtung und Umsetzungskompetenz. In diesem Zusammenhang lassen sich die gängigen Low-Code-Plattformen gemäss (Matvitsky, 2023) wie folgt kategorisieren, siehe Abbildung 3:

- **Challengers**
Oracle, Retool, HCLSoftware, Zoho
- **Leaders**
Mendix, OutSystems, Microsoft, ServiceNow, Salesforce, Appian
- **«Niche Player»**
Kintone, Unqork, Newgen und Huawei
- **Visionaries**
Creatio, Globant, Pegasystems

Abbildung 3: Magic Quadrant for Enterprise Low-Code Application Platforms (Matvitsky, 2023)

Andere Forschende wie Rymer und Koplowitz stufen die nachfolgenden Hersteller in ihren Anwendungsfeldern als führende Low-Code-Plattformen ein: *Appian, Kony, Mendix, Microsoft, OutSystems* oder *Salesforce* (Rymer & Koplowitz, 2019).

Low-Code definiert die Art des Programmierens, bei der Bausteine durch Drag-and-Drop miteinander kombiniert werden. Dadurch entsteht ein visueller Code, der eine individuell zusammengestellte Aktion ausführt. Der Begriff Low-Code umfasst jedoch verschiedene Ausrichtungen in Bezug auf die Bearbeitung des Codes. So sind die einen eher für erste Versuche, andere auf fortgeschrittene Entwickelnde ausgerichtet sind. Einen Überblick über diese Unterschiede bietet die nachfolgende Tabelle 1:

Tabelle 1; Vergleich verschiedene Low-Code-Plattformen

	LCDP	LCP	LCAP
Begriff-Hintergrund	Betont den gesamten Software-Entwicklungsprozess (Development) von der Modellierung bis Bereitstellung	Fokussiert sich auf die Bereitstellung einer Plattform für Rapid-Prototypen und einfache App-Erstellung	Heben den Anwendungsfokus (Applikation) hervor, also nicht nur Entwicklung, sondern Lifecycle und Betrieb.
Funktionen	End-to-End-Lifecycle: Requirements-Capture, Modell-to-Code-Generierung, Test, CI/CD-Integration Governance, Versionsverwaltung, ALM-Tools	Drag-and-Drop UI-Design, Formulare, einfache Datenanbindung Fokus auf Fachanwender und schnelle Prototypen	Vollständiges Anwendungsmanagement: Design, Depolymet, Monitoring, Updates integrierter Backend-Services (Daten, Auth, APIS)
Zielgruppe	Professionelle Entwickler und IT-Architekten, die agile Methoden mit automatischer Code-Generierung verstärken wollen.	Citizen Developers und Fachabteilungen, die ohne tiefes Coding schnell Lösungen bauen möchten	Mischteams aus Fachanwender und DevOPS/IT-Operationen: von Prototyp bis Live-Betrieb
Governance und Sicherheit	umfassende Rollen-, Release-Management und Security-Features	Grundlegende Zugriffskontrolle und einfach Depoly-Optionen	Erweiterte Policy-Verwaltung, Laufzeit-Monitoring, SLA- und Compliance-Funktionen
Anwendung	Enterprise-Fokus	Schneller Einstieg, business-zentriert	End-to-End App Delivery
Anbieter (nicht abschließend)	<i>Mendix, OutSystems, Pega, IBM Rapid Developer, Altova, MetaWizard</i>	<i>Microsoft Power Apps, Appian, Google AppSheet, BettyBlocks</i>	<i>Salesforce Lightning Platform, ServiceNow App Engine, MendixLCAP, Kissflow</i>

Die Low-Code Development Platform (LCDP)

Professionelle Entwickler und Architekten werden mit einer Low-Code Development Platform während des gesamten Software-Lebenszyklus unterstützt. Dies von der Anforderungserfassung über die modellbasierte Codegenerierung bis hin zu automatisierten Tests und integrierten Continuous Integration oder Continuous Delivery (CI/CD)-Pipelines. Die Plattform kombiniert erweiterte Governance- und Release-Management-Funktionen mit integrierten Application Lifecycle Management (ALM)-Werkzeugen. Dadurch lassen sich Versionierung, Rollbacks und Deployments auch in komplexen Unternehmensumgebungen effizient durchführen und kontinuierlich überwachen. So entsteht auf diese Weise aus abstrakten Fachmodellen innerhalb kurzer Zeit strukturierter und wartbarer Produktionscode. Teams können sich dadurch stärker auf fachliche Innovationen konzentrieren. Zudem ermöglicht die breite Einsetzbarkeit der Plattform, dass sowohl professionelle Softwareentwicklern bis hin zu nicht-technischen Fachexperten in ein Projekt eingebunden werden können (Overeem et al., 2021, p. 381).

Low-Code Platform (LCP)

Die Nutzung einer Low-Code-Plattform (LCP) ermöglicht es Fachanwendern und Citizen Developern auch ohne tiefgehende Programmierkenntnisse in kurzer Zeit funktionsfähige Anwendungen zu erstellen. Mithilfe einer Drag-and-Drop-Oberfläche entwerfen sie Benutzeroberflächen, binden über vorgefertigte Konnektoren Datenquellen an und definieren einfache Geschäftslogiken. Standardisierte Bereitstellungs-Optionen und Basis-Zugriffssteuerungen sorgen dafür, dass Lösungen schnell in Produktion gehen können, ohne die IT-Abteilung zu überlasten.

Dadurch lassen sich einfache Prototypen und Geschäfts-Apps innerhalb weniger Tage statt Wochen entwickeln und bei Bedarf schnell an neue Anforderungen anpassen. (Sahay et al., 2023)

Low-Code Application Platform (LCAP)

Bei LCAP handelt es sich um Anwendungsplattformen, die eine schnelle Entwicklung, Bereitstellung, Ausführung und Verwaltung von Anwendungen ermöglicht. Sie nutzen dazu deklarative und abstrakte Programmieransätze auf einer hohen Ebene, wie zum Beispiel modellgetriebene und metadatenbasierte Programmiersprachen sowie einstufige Bereitstellungen. Weiter unterstützen LCAP gängige Benutzerschnittstellen, Geschäftsprozesse und Datendienste. (Vincent et al., 2020, p. 1)

Application Programming Interfaces (API)

Die Entwicklung zu einem Ökosystem erfolgt, indem Teile der Anwendung über Programmierschnittstellen (APIs) eingebunden werden. Studien zeigen, dass Fragen zur Stabilität, Sicherheit und Skalierbarkeit von Ökosystemen eng mit den Funktionen des API-Managements verknüpft sind. Unter API-Management versteht man die Tätigkeit, mit der Organisationen ihre APIs entwerfen, veröffentlichen und bereitstellen, sodass auch (externe) Entwickler darauf zugreifen können. Dies ist eine der grundlegenden Praktiken für den Aufbau eines Ökosystems. (Overeem et al., 2021, p. 381)

2.1.2 Low-Coding vs. No-Coding

Beim Low-Coding wird der Code durch Ziehen und Ablegen der Komponenten entwickelt. Beim No-Coding werden ausschliesslich Bausteine kombiniert, welche den Code erzeugen. Der eigentliche Programmcode wird dabei im Hintergrund durch die Anwendung generiert (Adrian et al., 2020). Forrester Research unterscheidet zwischen Low-Code- und No-Code-Produkten gar anhand ihrer Zielgruppen. Low-Code-Anwendungsplattformen (LCAP) sind zwar breit gefächert und richten sich sowohl an professionelle Entwickler als auch an Hobbyentwickler, erfordern aber dennoch einige Programmierkenntnisse.

Daher richten sich Low-Code-Anwendungsplattformen in erster Linie an Vollzeit- und Teilzeitentwickler. No-Code-Produkte hingegen sind speziell auf Geschäftskunden ausgerichtet und ermöglichen die Entwicklung individueller Anwendungen ohne Programmierkenntnisse. (IBM, 2021)

Blockbasierte Programmierung «Blockly» (No-Code)

Dieser visuelle Code-Editor stammt von Google und verwendet grafische Blöcke, welche per Drag-and-Drop zusammengestellt werden können und teilweise der No-Code-Programmierung zugeordnet sind. Im Unterschied zu anderen Low-Code Anwendungen, ist hier der zugrunde liegende Code sichtbar und nachvollziehbar. Die farbigen Blöcke, welche mit Wortaufforderungen gefüllt sind und aus einer Auswahl zusammengestellt werden können, werden direkt in eine Programmiersprache nach Wahl, wie JavaScript oder Python, übersetzt und sind somit maschinenausführbar, siehe Abbildung 4. (Google, 2025)

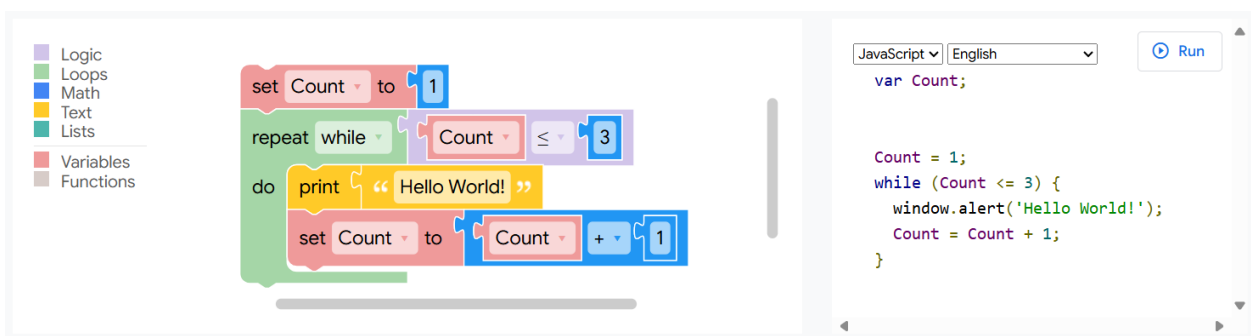


Abbildung 4; Beispiel Blockly-Programmierung (Google, 2025)

Der Fokus der blockbasierten Codierung liegt auf dem vereinfachten Einstieg von Anfängern aller Art in die Programmierung. So können bereits Schülerinnen und Schüler der Unterstufe erste Erfahrungen sammeln, was durch die grafische Puzzle-Teil-Optik zusätzlich unterstützt wird. Zudem ist Blockly als Open-Source-Software verfügbar, was die Nutzung im Bildungsbereich fördert.

2.1.3 IFC und Low-Code-Plattformen

Low-Code-Plattformen gibt es viele, jedoch sind viele nicht für die Arbeit mit BIM resp. IFC gebaut. In Tabelle 2 ist eine Zusammenstellung mit möglichen Anbietern sowie deren Eigenschaften in Kombination mit IFC.

Tabelle 2; Vergleich verschiedene Low-Code-Plattformen mit IFC

Plattform	IFC-Unterstützung	Beschreibung
Tririga (IBM)	Indirekt (über Integrationen)	Ist eine Low-Code-Plattform für das Facility Management, die eine Integration von BIM-Daten häufig über Forge oder eine andere Middleware ermöglicht. (IBM, 2022)
Autodesk Platform Services	Ja (vollständig)	Es handelt sich zwar nicht um eine klassische Low-Code-Plattform, jedoch stehen API sowie visuelle Workflows für die Integration und Verarbeitung von IFC zur Verfügung. Zudem können gewisse Low-Code-Tools damit kombiniert werden. (Autodesk Platform Services, 2025)
Mendix (Siemens)	Teilweise (über Add-ons / Java-Aktionen)	Mendix kann durch Erweiterungen IFC-Dateien verarbeiten, insbesondere durch die Integration von Siemens-eigenen Tools wie NX oder Teamcenter. (SIEMENS, 2024)
ShapeDiver (für Rhino/Grasshopper)	Ja (über Plug-ins)	Es handelt sich nicht um eine klassische Low-Code-Lösung, jedoch ermöglicht die Web-Plattform, 3D-Modelle mit visueller Programmierung zu parametrisieren und als IFC-Dateien zu exportieren. (ShapeDiver, 2025)
Speckle	Ja (über Konnektoren, Open Source)	Es handelt sich um eine Open-Source-Datenplattform für AEC, die visuellen Datenflüsse bereitstellt (ähnlich Low-Code) und IFC nativ unterstützt. (Speckle, 2025)
Power Platform (Microsoft)	Indirekt (über Custom Connectors)	Der Zugriff auf IFC-Daten ist über Power Automate oder Power Apps in Kombination mit API oder Middleware möglich.

Im Zusammenhang mit Nutzung vom IFC-Datenschema als Datenbasis wird nachfolgend auf die beiden Low-Code-Plattformen: Autodesk Platform Services (APS) und Microsoft Power Platform noch näher eingegangen

2.1.4 Autodesk Platform Services (APS)

Die Autodesk Platform Services (APS), früher bekannt als Autodesk Forge, ist eine Sammlung von Cloud-Anwendungsschnittstellen (API). Diese Schnittstellen ermöglichen die Nutzung von Konstruktions- und Projektdaten aus der Autodesk-Umgebungen (z.B. BIM 360, Autodesk Construction Cloud, Fusion Team) sowie eigenen Anwendungen wie z.B. Excel oder Structured Query Language (SQL)-Datenbanken.

Mit APS können Unternehmen ihre Arbeitsabläufe digital erweitern und ihre Prozesse zentral in einer Anwendungsoberfläche koordinieren.

- Die Data Management API ermöglicht den Zugriff auf Dateien, Versionen und Metadaten.
- Die Model Derivative API übersetzt CAD-/BIM-Dateien (z.B. IFC, Revit, DWG) in anzeigebare Formate und extrahiert deren Eigenschaften.
- Die Viewer-API erlaubt die interaktive Darstellung von 2D- und 3D-Modellen direkt im Web oder in mobilen Anwendungen.
- Die Design Automation API führt CAD-/BIM-Operationen serverseitig automatisiert aus.
- Mithilfe von Webhooks können Anwendungen auf Ereignisse wie Datei- oder Versionsänderungen reagieren und ermöglichen so eine ereignisgesteuerte Kommunikation zwischen den beteiligten Akteuren.

APS stellt damit grundlegende Bausteine bereit, um Dateien hochzuladen, zu verarbeiten, anzuzeigen und mit eigenen Applikationen zu verknüpfen. Auf dieser Basis lassen sich massgeschneiderte, cloudbasierte Workflows für Konstruktions- und Projektdaten entwickeln.

2.1.5 Microsoft Power Platform

Die Microsoft Power Platform gehört zu den am flächendeckendsten vertretenen Low-Code-Plattformen/No-Code-Tools. Ein grosser Vorteil besteht darin, dass in vielen Firmen bereits viele andere Microsoft-Programme wie: Outlook, Word oder Excel genutzt werden und sich somit sehr einfach in die Arbeitsprozesse einbinden lassen. Mit ihren Kernkomponenten, wie in der Abbildung 5 dargestellt, unterstützt die Microsoft Power Platform zahlreiche Optimierungen von Geschäftsprozessen.

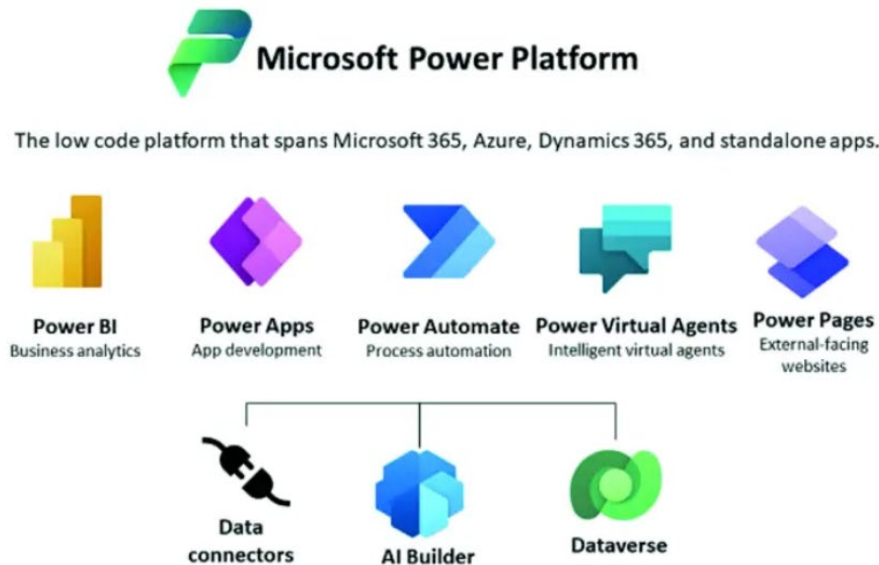


Abbildung 5; Kernkomponenten Microsoft Power Platform (Microsoft, 2025a)

Nachfolgend werden diese Komponenten kurz in ihren vereinfachten Funktionen und Eigenschaften erörtert:

Power BI

- **Datenanbindung:** Verbindet variabel viele Datenquellen. Zum Beispiel Excel, SQL-Datenbanken, Dynamics 365, Sharepoint und weitere
- **Dashboards und Berichte:** interaktive Visualisierung von Tabellen und Datenbanken in Echtzeit. (Microsoft, 2025b)

Power Apps

Grundsätzlich stellt PowerApps den Benutzer vor die Grundsätzliche Frage, welche Art von App er erstellen will:

- **Canvas-Apps:** Drag-and-drop-Oberfläche mit der das Layout der App frei gestaltet werden kann.
- **Model-Driven-Apps:** Automatisch generierte Apps auf der Basis von Datendefinitionen im Dataverse generierte Apps, sehr gut geeignet für komplexe Zusammenhänge. (Microsoft, 2025c)

Dabei bietet die Canvas-App grössere Freiheiten bei Quellen- oder Visualisierungsaspekten und ermöglicht individuellere Einstellungen.

Power Automate

- **Flows:** Alles, was zur täglichen Routine gehört automatisieren. Dies kann das Löschen von Doppelten Dokumenten aber auch das Auslesen von Daten sein.
- **Robotic Process Automation:** Automatisierung von Desktop- oder Web-aufgaben, die kein Web Application Programming Interface (Web-API) Zugang haben. (Microsoft, 2025d)

Power Virtual Agents

- **Chatbots:** selbst Erstellen von Chatbots ohne Programmieraufwand. (Aftab & AI, 2023)

Power Pages

- **Webseiten:** Erstellen von Websites als Kundenportalen, Intranetseiten und anderen Anwendungsfällen. (Microsoft, 2025e)

Microsoft Dataverse

- **Datenbank:** Speicherung und Verwaltung von Daten zur Nutzung in Geschäftsapplikationen (zentrale Datenbasis für Power Platform). (Microsoft, 2025a)

Microsoft AI builder

- **KI:** Bereitstellung vorgefertigter und anpassbarer KI-Modelle, die in eigene Anwendungen integriert werden können. (Microsoft, 2025f)

KI-Copilot

Der KI-Copilot unterstützt bei der Erstellung von Anwendungen, sowie der Verwaltung von Daten durch natürliche Sprache (Navin, 2023). Er ist sowohl in PowerApps, PowerAutomate, PowerBI aber auch in Power Virtual integriert.

Weitere Verknüpfung von Microsoft mit weiteren Programmen

Die in Abbildung 6 gezeigten Verknüpfungen sind nicht abschliessend und werden beinahe täglich erweitert.

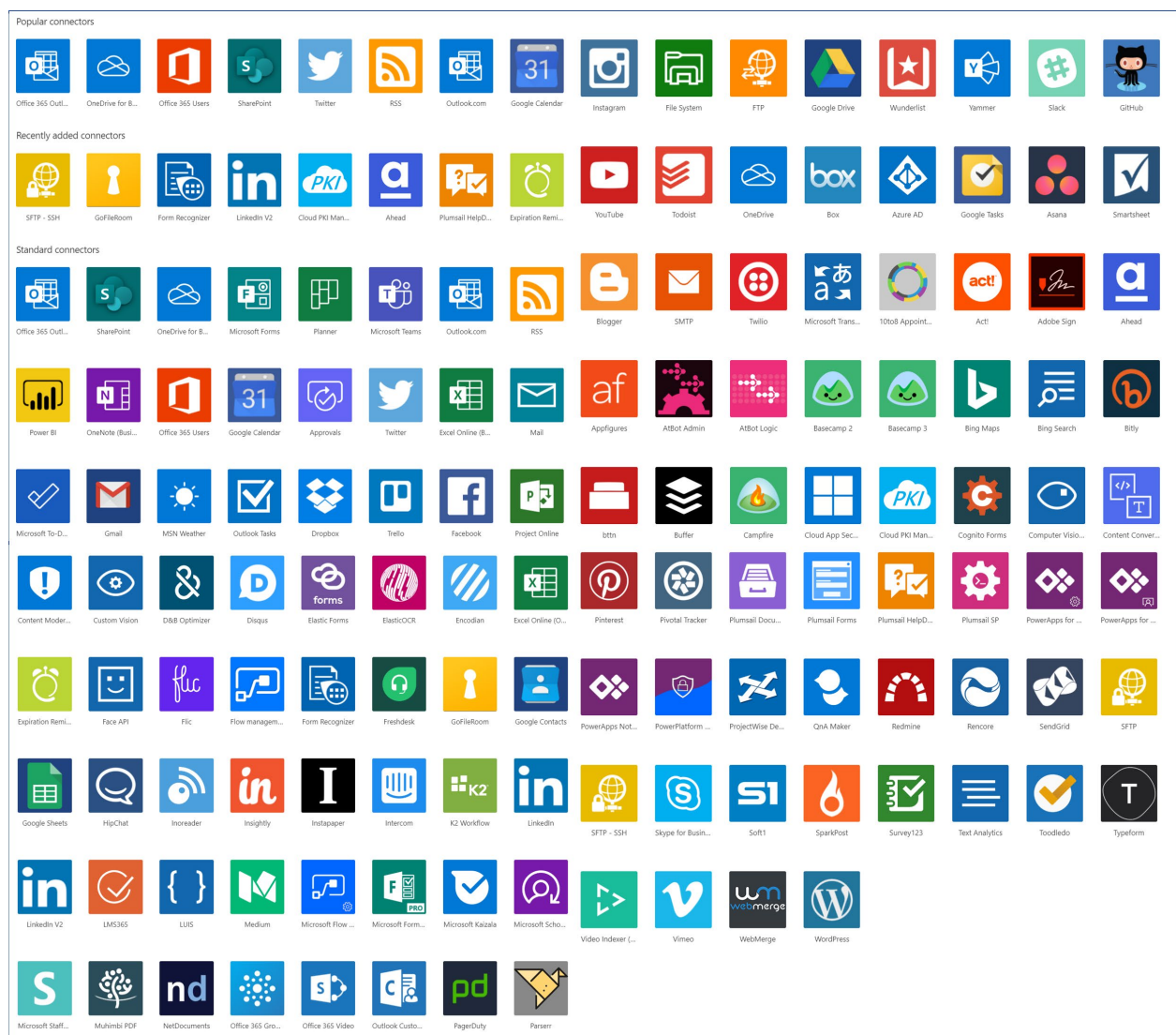


Abbildung 6; Power Automate Connectors (Tracy van der Schyff, 2020)

Web-API

Viele weitere Anbindungen sind über eine Web-API möglich. Dabei kommunizieren die Softwareanwendungen übers Web miteinander. Solche Einbindungen erfordern aber die Fertigkeit des Hardcodierens. Mit diesem System wäre es möglich die Open-Source-Datenplattform Speckle in die App einzubinden (Speckle, 2025).

3D-Viewer

In Microsoft PowerApps besteht die Möglichkeit, Inhalte aus Power-BI direkt zu integrieren. Damit lassen sich datenbasierte Entscheidungen innerhalb der App unterstützen. Hierzu kann beispielsweise mithilfe der zusätzlichen Applikation von Tracer ein Power BI Viewer in eine Power-App integriert werden.

Diese ermöglicht die Darstellung von digitalen Bauwerksmodellen und anderen IFC-basierenden Informationen mithilfe von Dashboards in Power BI innerhalb der App. Voraussetzung für die funktionsfähige Integration ist dabei ein veröffentlichtes Dashboard in Power BI. Nur so können die gewünschten Visualisierungen eines IFC-Modelles korrekt im Viewer dargestellt und genutzt werden. Die Zusatzapplikation Tracer wurde zuletzt im Jahr 2020 aktualisiert, was Fragen zur Aktualität und langfristigen Integration aufwirft. (Proving Ground, 2022).

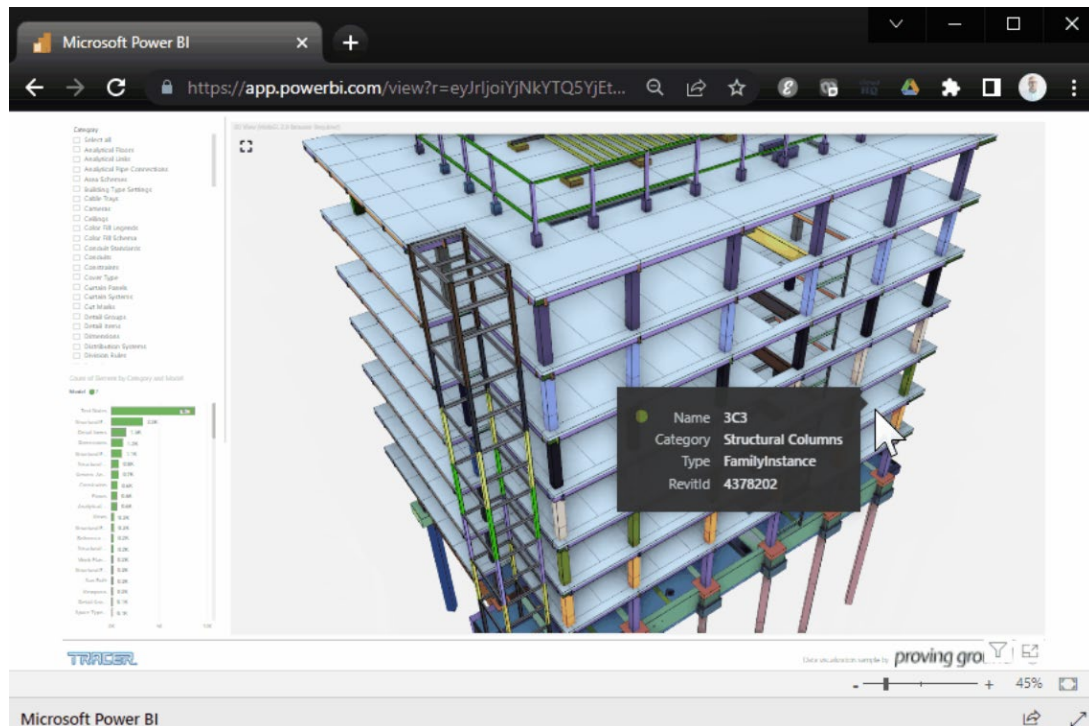


Abbildung 7; Tracer in der Umgebung von PowerBI (Proving Ground, 2020)

2.1.6 Model Driven Engineering (MDE)

Anstatt sich in der Softwareentwicklung auf den Code zu konzentrieren wird bei Model Driven Engineering (MDE) mit **Modellen** gearbeitet (Palacios-González et al., 2008, p. 73). Dieser Ansatz beginnt mit abstrakten Modellen, wie beispielsweise einem UML-Diagramm. Dabei kann die Definition des Modells wie folgt lauten:

«Ein Modell eines Systems ist eine Beschreibung oder Spezifikation dieses Systems und seiner Umgebung für einen bestimmten Zweck. Ein Modell wird oft als eine Kombination aus Zeichnungen und Text dargestellt. Der Text kann in einer Modellierungssprache oder in einer natürlichen Sprache verfasst sein.»
(Mukerji & Miller, 2003)

MDE ist ein Oberbegriff für die Konzepte: MDSE, MDA und MDD. Der zentrale Gedanke besteht darin, dass Entwickler grundlegende Zusammenhänge planen und verschiedene Modelle miteinander verknüpfen können. Dies geschieht dabei auf einem geringen Abstraktionsgrad (Palacios-González et al., 2008, p. 73). Dadurch wird die Lücke zwischen dem zu lösenden Problem und der Software-Implementierung geschlossen. Der Ansatz setzt dabei auf der semantischen Ebene an (France & Rumpe, 2007, p. 37).

Model Driven Software Engineering (MDSE)

Dabei handelt es sich um ein Synonym zu MDE, welches oft in wissenschaftlichen Publikationen verwendet wird. Im Model Driven Software Engineering (MDSE) nutzt man Low-Code- oder No-Code-Tools. Diese Werkzeuge helfen bei der Softwareentwicklung, indem sie Modelle als wichtigste Eingabe verwenden. Mit diesen Modellen kann Software automatisch erzeugt werden. Zu solchen Werkzeugen gehören modellgetriebene und modellbasierte Entwicklungsumgebungen sowie Low-Code- und No-Code-Plattformen (Mosquera et al., 2024, p. 1).

Model Driven Architecture (MDA)

Eine weitere Variante ist der Model Driven Architecture (MDA) Ansatz. Dieser beruht auf dem Spezifikationsstandard der Object Management Group, der im Jahr 2001 eingeführt wurde. Der Ansatz unterscheidet drei Abstraktionsebenen:

- Das Computation Independent Model (CIM) bildet ein fachliches Modell ohne technische Details.
- Das Platform Independent Model (PIM) beschreibt ein domänenorientiertes Systemmodell.
- Das Platform Specific Model (PSM) erweitert das PIM um platformsspezifische Aspekte.

Zudem werden formale Transformationsschritte von CIM über PIM zu PSM bis hin zum Code definiert. Dadurch wird die Interoperabilität zwischen verschiedenen Werkzeugen unterstützt. (Mukerji & Miller, 2003, p. 2 ff)

Model-Driven Development (MDD)

Alternativ ist weiter der Ausdruck Model-Driven Design (MDD) zu lesen. Er bezeichnet eine praxisorientierte Teilmenge von Model-Driven Engineering, mit Schwerpunkt auf automatischer Codegenerierung und schneller Iteration. Im Vordergrund steht der Einsatz von Low-Code- und No-Code-Plattformen, auf denen Modelle angepasst und mit minimalem Aufwand lauffähige Softwareprodukte erzeugt werden können. Der Begriff findet besonders in kommerziellen Produkten wie: Mendix MDD-Studio oder der Microsoft Power Platform Verwendung. (Ameller et al., 2015, p. 5)

Computer Integrated Construction (CIC)

Konkreter auf die Baubranche bezogen wird der Computer Integrated Construction (CIC) Ansatz beschrieben, bei dem die gesamte Wertschöpfungskette mithilfe von IT-Systemen und digitalen Prozessen durchgängig verknüpft wird. Dies betrifft alle Phasen, von der Planung über die Fertigung bis hin zur Montage. Das Ziel besteht darin, den Ablauf effizienter zu gestalten und gleichzeitig manuelle Schnittstellen sowie Medienbrüche weitgehend zu reduzieren. (Elzarka, 2017, p. 1)

2.2 Die verschiedenen Akteure / Beteiligten

Die Nutzenden von Low-Code reichen von professionellen Entwickelnden bis hin zu sogenannten Citizen Developern. Wie in der Abbildung 8 ersichtlich haben aber auch die die Unternehmen ihren Einfluss auf den Einsatz von Low-Code-Anwendungen in der Praxis.

Bei Citizen Developern handelt es sich um Personen, die wenig/keine professionelle Programmierausbildung besitzen und deshalb als Amateurprogrammierende gelten. Sie verfügen jedoch in der Regel über umfassendes Fachwissen in ihrem jeweiligen Bereich. Durch den Einsatz von Low-Code können Fachleute zu Citizen Developern werden und so ihr Fachwissen zusammen mit der unerschwelligen Programmierung prozessoptimierend einbringen. Gleichzeitig sollten Low-Code-Plattformen so gestaltet sein, dass sie auch professionelle Entwickelnde mit einer formalen Ausbildung, in ihrer Produktivität unterstützen. (Hirzel, 2023, p. 76)

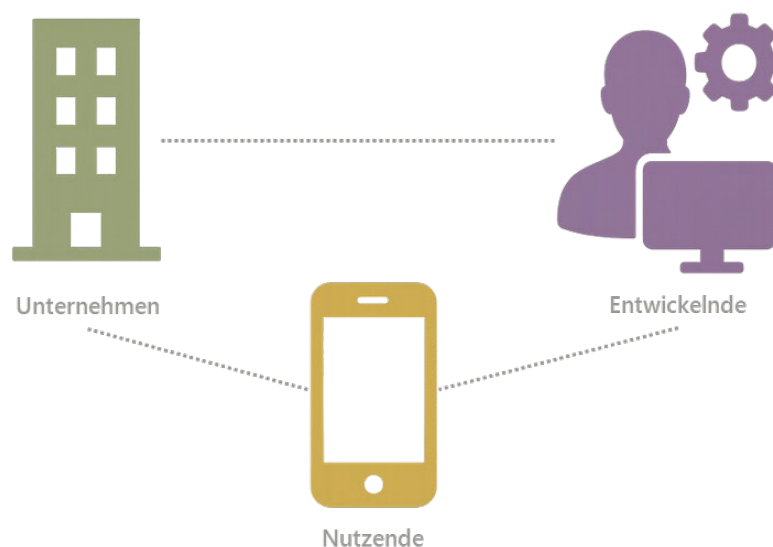


Abbildung 8; die verschiedenen Beteiligten (eigene Abbildung)

2.2.1 Perspektive Unternehmen



Abbildung 9; Die Unternehmung
(eigene Darstellung)

Unternehmen können unterschiedliche Beweggründe für die Einführung von Low-Code-Plattformen haben. Ein zentraler Faktor ist der Mangel an professionellen Entwicklenden, die in der stark zunehmenden digitalisierten Welt stark nachgefragt sind.

Hinzu kommt, dass bei mühsamen manuellen Aufgaben eine bedeutende Fehlerquelle entstehen kann, die es zu verringern gilt. Gleichzeitig kann der Einsatz von Low-Code-Programmen zu einer Zeitersparnis führen, welche wieder mehr für wertschöpfende Tätigkeiten von Angestellten genutzt werden kann. (Hirzel, 2023, p. 76)

2.2.2 Perspektive Entwickelnde



Abbildung 10; Die Entwickelnden
(eigene Darstellung)

Auch bei Entwicklenden ist die Low-Code-Technologie ein wachsender Entwicklungsansatz, der von vielen Plattformen unterstützt wird. Er schliesst die Lücke zwischen Business und IT, indem er die aktive Mitarbeit von: IT-Fachkräften und nicht-technischen Fachexperten, den sogenannten Citizen Developern, im gesamten Lebenszyklus der Anwendungsentwicklung unterstützt. (Khorram et al., 2020).

IT-Fachkräfte und Low-Code-Developer

Nach Stople et al. können die IT-Fachkräfte anhand ihrer Themenfelder, Anwendungstechnologien und Zuständigkeiten unterschieden werden. Deren Merkmale sind in der Tabelle 3 dargestellt und beziehen sich auf die erfahrenen IT-Fachkräfte (Heavyweight IT) sowie auf die weniger erfahrenen IT-Fachkräfte (Lightweight IT). (Stople et al., 2017)

Tabelle 3; versierte IT-Fachkräfte und ungeübte IT-Fachkräfte (Stople et al., 2017, p. 5)

	Heavyweight IT	Lightweight IT
Profile	Back-end: Supporting documentation of work	Front-end: Supporting work processes
Systems	Transaction systems	Process support, apps, BI
Technology	Servers, databases, enterprise bus technology	Tablets, electronic whiteboards, mobile phones
IT architecture	Centralized or distributed	Meshworks
Owner	IT department	Users and vendors
Development culture	Systematics, quality, security	Innovation, experimentation
Problems	Increasing complexity, rising costs	Isolated gadgets, security
Discourse	Software engineering	Business innovation

Die Rolle von IT-Fachleuten im Kontext von Low-Code-Technologien wurde in der bisherigen Literatur kaum beleuchtet. In vielen Studien steht der potenzielle Nutzen für Fachabteilungen und Citizen Developer im Vordergrund. Die Auswirkungen auf professionelle IT-Fachkräfte werden hingegen nur selten thematisiert. Noack & Fedtke beschreiben in ihrer Publikation, dass es sich bei einem Low-Code-Developer um einen professionellen Softwareentwickler handeln muss, welcher seine Codes folglich mit dieser Technologie erstellt (Noack & Fedtke, 2025, p. 62).

Citizen Developer

Als Citizen Developer werden Personen innerhalb eines Unternehmens bezeichnet, die Softwareanwendungen, Tools oder Workflows erstellen, ohne eine formale Ausbildung in der traditionellen Softwareentwicklung zu haben. Power-User, also Personen mit Fachwissen und technischen Kenntnissen, gelten dabei als besonders qualifizierte

Citizen Developer. Ihr Fachwissen verbindet die Anforderungen des Unternehmens mit der IT-Abteilung und ermöglicht es, Probleme mithilfe von Low-Code-Plattformen rasch zu lösen. Mit der Befähigung von Power-Usern wird die Flexibilität innerhalb der Organisation zusätzlich gesteigert und gleichzeitig nimmt die Abhängigkeit von IT-Abteilungen ab. Navin beschreibt weiter, dass auch die Eigenverantwortung bei der Erstellung von Softwarelösungen und Arbeitsablaufoptimierung in Unternehmen steigt. (Navin, 2023)

2.2.3 Perspektive Nutzende



Abbildung 11: Die Nutzenden (eigene Darstellung)

Am Schluss der Erstellung einer Applikation stehen die Nutzenden. In der Literatur werden diese als solches leider nicht weiter beschrieben. Es fehlt sowohl eine eindeutige Definition als auch eine Darstellung ihrer Erwartungen. Das erschwert den Prozess der Feinabstimmung und die Gestaltung der Benutzerinteraktion im Unternehmen. (Mosquera et al., 2024, p. 16)

Grundsätzlich können Nutzende aber alle sein, also Unternehmensinterne und auch Externe. In dieser Arbeit wird diese Akteursgruppe nicht weiter eingegrenzt, da eine solche Applikation grundsätzlich für Alle hilfreich sein sollte.

Bauleitung

Ein Akteur der Nutzenden in der Bauindustrie, welcher potenziell die Low-Code-Technologie nutzen könnte, sind die Bauleitenden. Laut Duden sowie im allgemeinen Sprachgebrauch ist die Bauleitung mit der Führung und Überwachung der Ausführung von Bauten beauftragt (Duden, 2025). Dazu gehören die Koordination, Anordnung, Instruktion und Kontrolle sämtlicher Bauarbeiten auf der Baustelle. Im Wesentlichen umfasst dies die Umsetzung architektonischer und planerischer Vorgaben in die bauliche Realität. Diese Aufgabe lässt sich auf die drei Kernkompetenzen: Zeit, Kosten und Qualität zurückführen. Die Bauleitung stellt das Bindeglied zwischen Planung und Ausführung dar und übernimmt damit die zentrale Kommunikationsschnittstelle zwischen beiden Bereichen (SIA 102, 2020, pp. 11, 28).

2.3 Gesetzliche und Normative Rahmenbedingungen

Schnelle und Agile Entwicklungen einer Applikation mit möglichst kleinem Aufwand sind auf gesetzliche und normative Grundlagen als Basis angewiesen. Konkret sind diese Grundlagen für Bauleistungen während der Ausführungsphase in der SIA 102 Leistung und Honorare der Architektinnen und Architekten geregelt (SIA 102, 2020, p. 28).

2.3.1 Baujournal

In der Praxis ist die Bauleitung verpflichtet, ein Baujournal und daraus resultierend eine Pendenzenliste zu führen (SIA 102, 2020, p. 29). Das Baujournal ist zwar nicht in allen Einzelheiten vorgeschrieben, enthält jedoch in der Regel folgende wesentliche Inhalte (Menz, 2020):

Beschrieb und Ort des Bauvorhabens, Datum, Uhrzeit, Wetter, Temperatur und Luftfeuchtigkeit des Aufzeichnungstags, Anwesende (Arbeitsgattungen mit Anzahl der Arbeiter), Baufortschritt und Verzögerung allgemein und im Detail, Mängel und Abnahmen, Prüfungen und Messungen, besondere Vorkommnisse, Besprechungen und deren Teilnehmer, Entscheidungen, Vereinbarungen und Ergebnisse, Terminfestlegungen und Terminänderungen, Fotografische Dokumentation

Weiter sind im Internet auch Mustervorlagen zugänglich, Anhang C (Schweizerische Eidgenossenschaft, 2015). Darin sind auch die Allgemeine Bedingungen für Bauarbeiten geregelt. Im Bauprogramm wird der zeitliche Ablauf der Arbeiten innerhalb der vertraglich festgelegten Fristen definiert. So auch die Abgabe der Baujournals von ausführenden Gewerken an die Bauleitenden (SIA 118, 2013, p. 28).

2.3.2 Baufortschritt

Der Baufortschritt gibt Auskunft über den aktuellen Stand der Bauarbeiten eines Projekts. Er beschreibt, welche Bauphasen erreicht wurden und wie weit die Arbeiten fortgeschritten sind.

Im Ausland wird der Baufortschritt meist in Prozent angegeben. Dies hat einerseits damit zu tun, dass ein solcher

Wert greifbar ist und im Zusammenhang mit der geforderten Bezahlung des Unternehmers zu tun hat. So kann in Deutschland der Baumeister beispielsweise bei Beginn der Erdarbeiten bereits 30% der Auftragssumme in Rechnung stellen (Makler- und Bauträgerverordnung - MaBV, 2023).

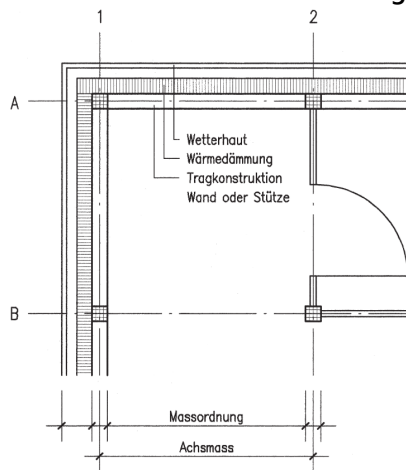
In der Schweiz werden Baufortschritte in Phasen und Teilphasen angegeben. Die Phasen gliedern sich in Vorstudie, Projektierung Ausschreibung, Realisierung und Bewirtschaftung. Die Realisierung gliedert sich nach SIA wiederum in folgende Teilphasen (SIA 112, 2017, pp. 7–8):

- **Ausführungsprojekt** - Detailpläne, Ausführungsunterlagen für den Unternehmer
- **Ausführung** - Bauarbeiten, Bauleitung, Qualitätskontrolle
- **Inbetriebnahme und Abschluss** - Funktionstests, Übergabe an Bauherrschaft

Zusätzlich wird in der Ausführungsphase von Hochbauten weiter in folgende Fertigstellungs-Status unterschieden:

- Baubeginn und Arbeitsvorbereitung
- Rohbau, Ausbau und technische Installation
- Fertigstellung
- Mängelrüge, Nachbesserung und Abnahme
- Vertragsabschluss

2.3.3 Räumliche Strukturierung des Projektes



Zur systematischen Strukturierung und Orientierung in Grundrissen von Gebäuden werden in der Regel Achsen verwendet. Diese verlaufen in X- und Y-Richtung und bilden ein kartesisches Raster, das der räumlichen Gliederung dient. Die Achsen werden üblicherweise nummeriert (zum Beispiel 1, 2, 3 ...) bzw. mit Buchstaben bezeichnet (A, B, C ...) und dienen als Referenz für die Positionierung von Bauteilen wie Wänden, Stützen, Decken oder Öffnungen.

Gebäude können mittels Achsen strukturiert und in meist gleich grosse Bereiche unterteilt werden. Diese Massordnung für die Organisation von Grundrissen kann mittels Haupt- und Nebenachsen definiert werden. Dieser Raster setzt sich aus meist senkrecht zueinanderstehenden und parallelen Achsen zusammen. Dies geschieht meist mittel Axialachsen, welche sich in Haupt oder Nebenachsen gliedern lassen. (SIA 400, 2000, p. 18)

Abbildung 12; Beispiel einer Anwendung des Axialrasters (SIA 400, p. 18)

Die Struktur nach Gebäuden und Geschossen kennt man vor allem aus dem Hochbau. Für Bauprojekte im Tiefbau kommen hingegen oft Haupt- oder Längsachsen zum Einsatz.

2.4 Projektplanung auf der Baustelle

Um den Stand der Bauarbeiten messen zu können, muss vorgängig eine Planung des Bauablaufes stattfinden. Für diese Projektplanung können Hilfsmittel wie das Gantt-Diagramm eingesetzt werden. Dieses stellt den zeitlichen Ablauf von Aufgaben in Form eines Balkendiagramms dar. Dabei werden im Wesentlichen folgende Informationen dargestellt oder siehe Abbildung 13:

- **Aufgaben/Aktivitäten** des Projektes auf der vertikalen Achse.
- **Zeitverlauf** auf der Horizontalen Achse wie beispielsweise Tage, Wochen, Monate.
- Balken visualisieren die **Dauer einer Tätigkeit** mit Start- und End-Zeitpunkt. (Gantt, 1910)

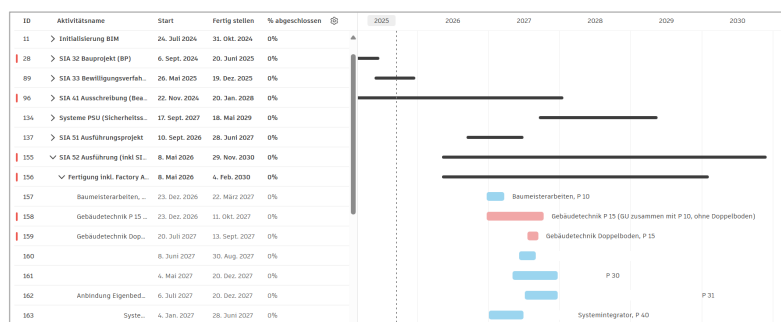


Abbildung 13; Auszug aus einer Projektplanung in ACC (eigene Abbildung)

2.4.1 Last Planner System (Aufgaben / Aktivität)

Das Last Planner System (LPS) ist eine kollaborative Planungsmethode aus dem Lean Construction-Ansatz. Diese zielt darauf ab, die Zuverlässigkeit und Effizienz auf Baustellen zu erhöhen, indem es eine standardisierte, transparente und effiziente Planung ermöglicht. Im Zentrum steht die Einbindung der ausführenden Personen, also derjenigen, die die Arbeit tatsächlich umsetzen, die sogenannten «Last Planner». Durch eine strukturierte Planung in mehreren Phasen (z.B. Phasenplanung, Wochenplanung und tägliche Abstimmung) wird sichergestellt, dass nur realistisch umsetzbare Aufgaben in den Plan aufgenommen werden. Dadurch werden Störungen reduziert, die Termintreue verbessert und ein kontinuierlicher Verbesserungsprozess etabliert.

Dieses System wird grösstenteils bei der Ausführung auf der Baustelle zur Anwendung. Auf die Berufsgattungen bezogen sind das meist Poliere und Montageleiter, welche in sehr grosses Interesse an Verbindlichkeit und Zuverlässigkeit haben um ihre Arbeitsabläufe sehr effizient umzusetzen.

Mit Hilfe einer unabhängigen Moderation werden bei der Methodenanwendung die nachfolgenden fünf Schritte vorgenommen:

1. Analyse und Optimierung des gesamten Prozesses
2. Festlegung der wichtigsten Meilensteine
3. Planung der nächsten Wochen
4. Gemeinsame detaillierte Besprechung der Arbeiten für die nächste Arbeitswoche:
Sind alle Arbeiten klar und für die Realisierung bereit?
5. Beurteilung der Arbeiten der letzten Woche (adaptiert von Scherer et al., 2021, pp. 80–81)

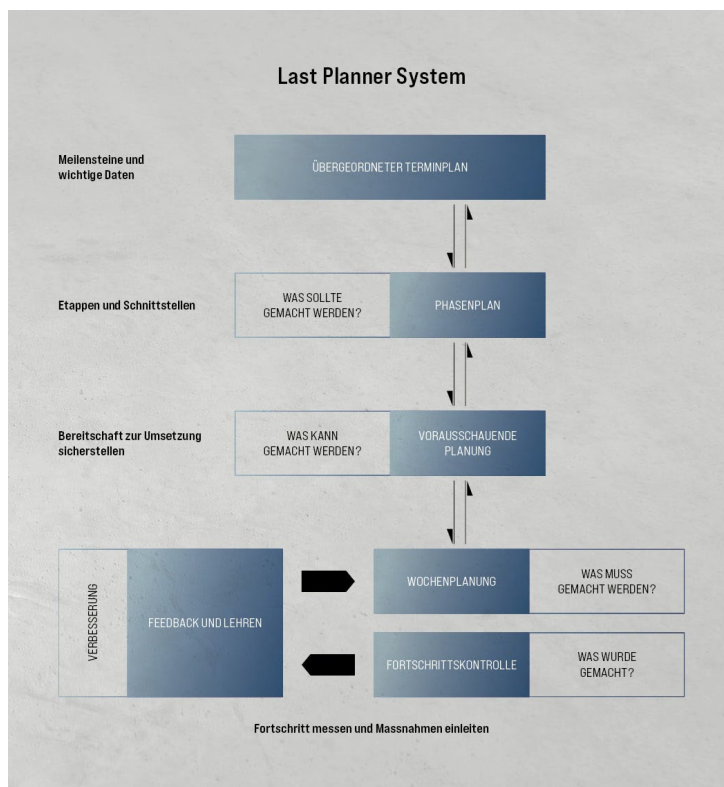


Abbildung 14; Last Planner System (adaptiert von Scherer et al., 2021, pp. 80–81)

Dabei ist das Last Planner-Produktionskontrollsystem eine Kombination aus: Philosophie, Regeln und Verfahren sowie eine Reihe von Werkzeugen, die die Umsetzung dieser Verfahren erleichtern. Im Verfahren selbst ist:

Einerseits die Kontrolle der Produktionseinheiten zentral. Durch fortlaufendes Lernen und den Einsatz von Korrekturmaßnahmen wird die Verteilung der Aufgaben schrittweise verbessert, sodass die Mitarbeitenden gezielt angeleitet werden können.

Andererseits hat die Steuerung des Arbeitsflusses einen grossen Einfluss aufs Arbeitsergebnis. Bereits der Begriff weist darauf hin, dass sie vorausschauend Steuerung gewährleisten soll, dass die Arbeit in einer sinnvollen Reihenfolge und mit angemessenem Tempo durch die Produktionseinheiten geführt wird. (Ballard, 2000, p. 2)

Messbar kann das mit dem Prozentsatz der Planerfüllung (PPC) sein. Dieser ergibt sich aus dem Verhältnis der abgeschlossenen geplanten Aktivitäten zur Gesamtzahl der geplanten Aktivitäten und wird in Prozent angegeben und zeigt dadurch wie hoch die Produktivität ist. So bedeutet ein höherer PPC eine höhere Produktivität und einen besseren Fortschritt. (Ballard, 2000, p. 4)

Die Ursachen für eine schlechte Planqualität oder die Nichtdurchführung geplanter Arbeiten können auf jeder Organisationsebene, in jedem Prozess oder jeder Funktion zu finden sein. Gründe dafür könnten sein:

- Fehlerhafte Anweisungen oder Informationen, die dem Last Planner gegeben wurden; z.B. hat das Informationssystem fälschlicherweise angezeigt, dass Informationen verfügbar waren oder dass die erforderlichen Vorarbeiten abgeschlossen waren.
- Nichtanwendung von Qualitätskriterien bei der Zuweisung von Aufgaben, z.B. weil zu viel Arbeit geplant wurde.
- Mangelnde Koordination bei der gemeinsamen Nutzung von Ressourcen, z.B. weil ein Computer oder Plotter fehlte.
- Änderung der Prioritäten, z.B. weil Mitarbeiter vorübergehend für eine «dringende» Aufgabe abgezogen wurden.
- Entdeckte Konstruktions- oder Lieferantenfehler beim Versuch, eine geplante Aktivität durchzuführen.

Einerseits geht es um die klare Definition der jeweiligen Aktivität. Zudem ist eine Analyse der bestehenden Einschränkungen erforderlich. Schliesslich umfasst der Prozess auch das Abrufen von Arbeiten aus vorgelagerten Produktionseinheiten sowie die Abstimmung von Auslastung und Kapazität. (Ballard, 2000, p. 5)

2.4.2 Produktions- und Ablauforganisation

Als Voraussetzung für eine erfolgreiche Baustellenplanung gilt ein vernetztes Verständnis der auszuführenden Arbeiten sowie der Wechselwirkungen zwischen den einzelnen Arbeitsschritten. Auf einige produktions- und ablauforganisatorische Aspekte wird nachfolgend ausführlicher eingegangen.

Losgrösse

Ein gängiges Mittel um Baustellen und Phasen zu strukturieren ist die Bündelung von Arbeitsgattungen zu einem Los. So können beispielsweise der Schreiner, der Gipser und der Elektriker im gleichen Los zusammengefasst werden und von Bereich zu Bereich wechseln. Wesentlich ist, dass mehrere Arbeitsgattungen zusammen ein Los bilden. Der Logistik-Softwarehersteller proLogistik beschreibt die Losgrösse auf seiner Website wie folgt:

«Sie bezeichnet die Menge von Produkten oder Teilen, die direkt hintereinander ohne eine Unterbrechung der Fertigung produziert wird z.B. 5.000 Schrauben. Jede Umrüstung der Maschinen verursacht somit Rüstkosten und -zeiten, da während der Umrüstung nicht produziert werden kann. Aus dem Grund wäre es sinnvoll grosse Lose herzustellen. Daraus resultieren jedoch hohe Lagerbestände, die bis zu ihrer Auslieferung eingelagert werden müssen und Lagerkosten mit sich bringen.»
(proLogistik Group, 2025)

Somit hat der Schreiner als Beispiel für sein Los mit allen Türen fünf Tage. Der Gipser aber drei Tage für das gleiche Los, wobei beim Gipser aber zusätzlich eine Trocknungszeit von zwei Tagen dazukommt. Dabei ist die Grösse des jeweiligen Loses und die damit verbundene Planung der Anzahl der Arbeitenden, sowie der Fabrikationsort relevant für die abgestimmte Termineinhaltung.

Produktionsmethode

One-Piece-Flow oder Single-Piece-Flow bezeichnet die Herstellungsart eines Bauteils. Konkreter wird bei der Bearbeitung ein Arbeitsschritt nach dem Nächsten ausgeführt (Haghsheno & Wachter, 2019). So wird beispielsweise eine Türe vom Zuschnitt bis zur Oberfläche in einem kontinuierlichen Ablauf gefertigt. Dies ist sinnvoll, weil nur ein minimaler Aufwand für den internen Transport sowie die Lagerung unfertiger Elemente entsteht (Wiendahl, 2014, p. 45)

Dies setzt jedoch ein tiefes Fachwissen über die jeweiligen Arbeitsprozesse voraus, um die Bearbeitungszeiten vollständig abschätzen zu können. Nur unter dieser Bedingung ist es möglich, dass die einzelnen Gewerke innerhalb desselben Loses zwischen verschiedenen Bereichen zirkulieren können (Scherer et al., 2021, p. 78)

Bauetappen

Um den Ablauf eines Projekts besser zu strukturieren und die einzelnen Arbeitsschritte effizienter zu organisieren, werden Baustellen gemäss SIA112 in Bauetappen oder Bauabschnitte unterteilt. Durch diese Unterteilung lassen sich komplexe Bauvorhaben in überschaubare Einheiten gliedern, die nacheinander oder parallel ausgeführt werden können. So behalten alle Projektbeteiligte den Überblick über Fortschritte, Ressourcen und Termine und können mögliche Abhängigkeiten zwischen den Gewerken klarer erkennen und gezielter steuern. (SIA 112, 2017, p. 18)

Baustrukturierung

Ein Gebäude wird während der Bauphase zur besseren Orientierung in verschiedene Bereiche unterteilt. Üblicherweise erfolgt die Gliederung zunächst in das gesamte Gebäude und in die einzelnen Geschosse. Zusätzlich werden meist auch Treppenhäuser und Wohnungen separat erfasst. In einer noch feineren Gliederung erfolgt die Unterteilung in Räume und schliesslich in die einzelnen Bauteile. Die klare Zuordnung zum jeweiligen Bauteil, also die klare Bauteillokalisierung erfolgt mit dem Bauteilnamen und der Grösse. Das Hauptziel dieser Strukturierung muss immer darin bestehen, den grössten Mehrwert zu erzielen. (Scherer et al., 2021, pp. 76–79)

2.4.3 Baufortschritt im digitalen Bauwerksmodell (4D-BIM)

Ein zentrales Merkmal von 4D-BIM in der Bauplanung ist die Verknüpfung des 3D-Gebäudemodells mit zeitlichen Informationen (Borrmann et al., 2021, p. 400). Dadurch kann der zeitliche Ablauf des Bauprozesses direkt auf Basis des digitalen Modells abgebildet werden. Anders als bei traditionellen Planungsmethoden, bei denen der Bauzeitenplan, etwa in Form eines Balkendiagramms und das geometrische Modell getrennt voneinander behandelt werden, vereint 4D-BIM diese beiden Komponenten. Jedes Objekt im Modell wie beispielsweise eine Wand, eine Decke oder ein Fundament wird mit terminbezogenen Attributen versehen. Die Nutzung des 4D-Modells erfordert einen gegliederten Terminplan in Form eines Balkendiagramms. Dieses Diagramm ist auch im gewünschten Detaillierungsgrad nach Bauabschnitten gegliedert. Dies ist nötig, um die Arbeitsgattungen in ihrer sequenziellen Abfolge nachvollziehbar darstellen zu können. (Borrmann et al., 2021, p. 400)

Dazu zählen neben den Terminen auch Informationen zum vor- und nachgelagerten Prozessschritt (Scherer et al., 2021, p. 33). Die Termininformationen können im offen dokumentierten Industry Foundation Classes (IFC)-Datenschema einfach und nachvollziehbar als Attribute definiert werden. So kann der geplante Baubeginn (ScheduleStart), das geplante Bauende (ScheduleFinish) sowie die daraus resultierende Dauer (ScheduleDuration) erfasst werden. Optional lassen sich auch der tatsächliche Beginn (ActualStart) und das tatsächliche Bauende (ActualFinish) angeben. Zudem kann der Bearbeitungsfortschritt eines Bauelements durch das Attribut (Completion) abgebildet werden.

Diese terminlichen Informationen werden entweder manuell gepflegt oder aus externen Projektmanagement-Tools übernommen und in das BIM-Modell integriert, häufig über Schnittstellen mit standardisierte Datenformaten wie IFC oder CSV. Durch diese zeitliche Anreicherung des Bauwerksmodells wird der Bauprozess im digitalen Raum erlebbar. Die Bauabfolge kann beispielsweise als animierter Simulationslauf dargestellt werden, bei dem die Bauelemente in der Reihenfolge ihres Einbaus nacheinander eingeblendet werden können.

Auf diese Weise lassen sich räumlich-zeitliche Abhängigkeiten frühzeitig erkennen und beurteilen. So kann etwa überprüft werden, ob die Arbeiten in der korrekten logischen Reihenfolge vorgesehen sind und ob zeitliche Überschneidungen oder Behinderungen auftreten. Gerade bei komplexen Bauprojekten mit vielen beteiligten Gewerken stellt diese Form der Planung ein wertvolles Instrument dar, um Koordination und Ablauf zu verbessern. 4D-BIM unterstützt nicht nur die traditionelle Ablaufplanung, es fördert auch moderne Bauprozessabwicklungsmethoden wie Lean Construction oder Taktplanung.

Schliesslich bietet 4D-BIM auch Vorteile im Hinblick auf die Bauausführung, etwa bei der logistischen Planung, der zeitgenauen Bereitstellung von Materialien und Ressourcen sowie der Baufortschrittskontrolle. Änderungen im Zeitplan wirken sich unmittelbar auf das Modell aus und sind somit direkt optisch ersichtlich. Heute können solche Anpassungen aber nur mit einer kostenintensiven Autorensoftware durchgeführt werden. Für ausführende Bauunternehmen wäre es sehr interessant, wenn wiederkehrende Rückmeldungen zu den ausgeführten Arbeiten direkt im digitalen Modell über eine Low-Code-Applikation erfasst werden könnten. Martinez erörtert, dass es sinnvoll wäre, den Baufortschritt über einen Wert abzufragen und zu bearbeiten und gleichzeitig das entsprechende Bauteil zur Unterstützung visuell darzustellen. (Martinez, 2024b).

2.4.4 Applikationen zur Dokumentation vom Baufortschritt

Auf dem Markt gibt es bereits eine Applikationen und Softwares, mit welchen der Baufortschritt direkt auf der Baustelle dokumentiert werden kann.

SmapOne

Die in Abbildung 15 ersichtliche Applikation stammt vom Hersteller smapOne AG, welche eine Reihe von Baustellenapplikationen anbietet. Diese spezifische ist dabei für Bauarbeiter und Bauleiter konzipiert, wobei die Nutzenden den Baufortschritt direkt über ihre Smartphones und Tablets dokumentieren können. Damit können viel Details bezüglich: Bauaktivität, Materialien, Maschinen, Anlagen, Arbeitsbedingungen und Problemfälle ohne Medienbruch in eine Quelldatenbank überführt werden. (smapOne, 2025)

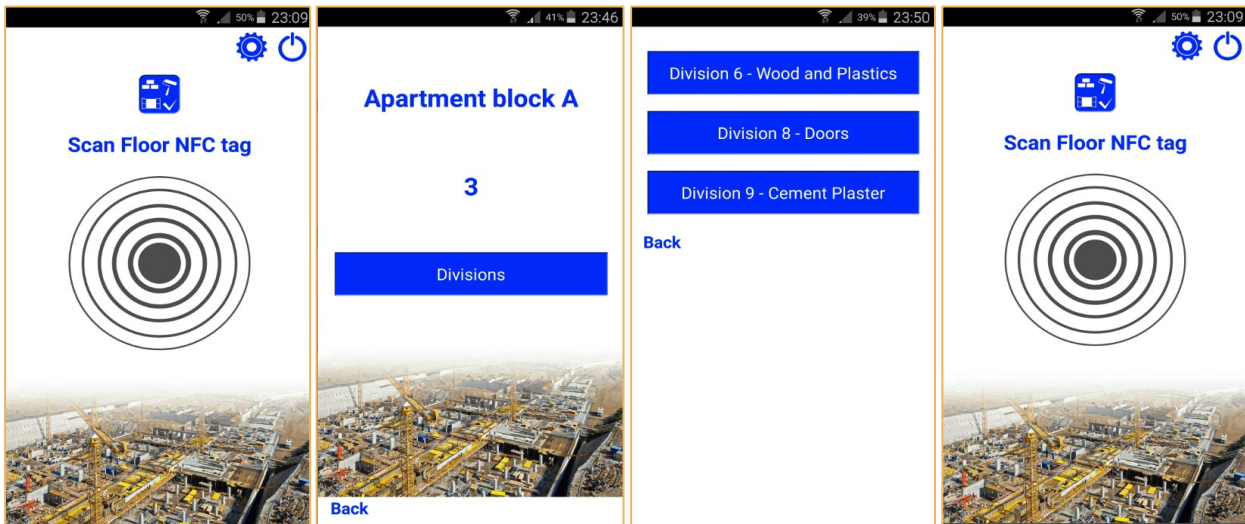


Abbildung 15; Bildschirmausschnitte der Low-Code-Applikation zur Dokumentation vom Baufortschritt (eigene Collage auf der Basis von smapOne Version 10.2, 08.11.2024)

So lassen sich in der Version 10.2 beispielsweise nachfolgende, nicht abschliessende Module nutzen:

- Datensatzauswahl
- Datums- und Zeitauswahl
- Foto- und Tonaufnahmen
- E-Mail-Eingabefeld
- Unterschrift und Zeitstempel

Messerli Protokoll

Bei dieser Applikation wie in der Abbildung 16 ersichtlich, handelt es sich um ein Modul der in der Schweiz bekannten Bauadministrationssoftware MesserliBAUAD vom Hersteller Messerli Informatik AG. Die Software ist ein geschlossenes System, in welchem unterschiedlichste Projekttereignisse dokumentiert werden können.

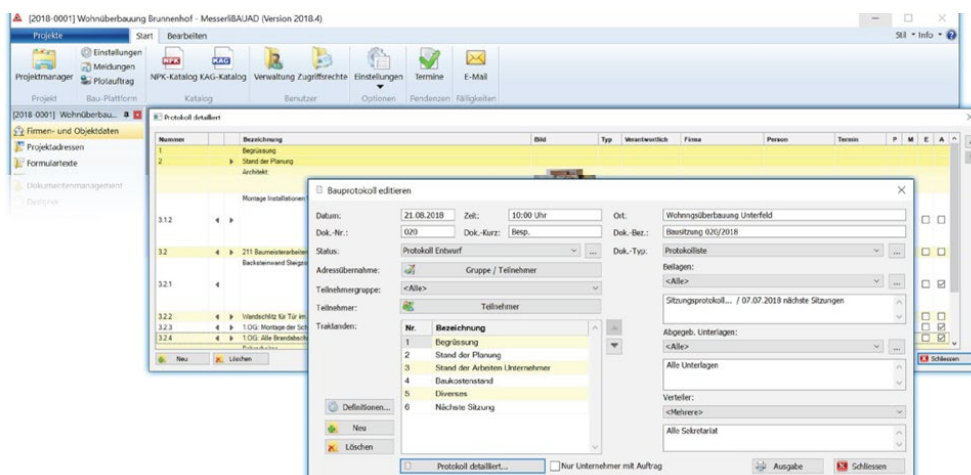


Abbildung 16; Ansicht zum Editieren eines Bauprotokolls (Messerli Informatik, 2021)

Diese Protokolle können dabei mit Informationen wie beispielsweise den Handwerkeradressen oder genutzten Materialien aus der Stammdatenbank ergänzt werden. Die erstellten Sitzungstraktanden dienen als Grundlage fürs Sitzungsprotokoll, wobei Pendenzen und weiterführende Informationen in Form von Entscheiden erfasst und an alle beteiligten Personen versendet werden können. Dieses System setzt jedoch eine konsequente Nutzung voraus und ist für externe Projektpartner nicht nutz- und einsehbar. Parallel dazu bleibt das digitale Bauwerksmodell von diesen Abläufen unberührt und wird nicht in den Prozess integriert. (Messerli Informatik, 2021, p. 23)

2.5 IFC-Datenschema

Das Industry Foundation Classes (IFC)-Datenschema ist ein international normierter und anerkannter Standard zur digitalen Beschreibung von Bauwerksinformationen über den gesamten Lebenszyklus eines Gebäudes (SN EN ISO 16739-1, 2024). Dieses ermöglicht es Informationen über das Bauwerk zu erfassen, zu verwalten und auszutauschen. Es wurde entwickelt, um den standardisierten Austausch und die Interoperabilität zwischen verschiedenen Softwarelösungen mit der Building Information Modeling (BIM) Methodik zu ermöglichen. Die offen dokumentierte Struktur folgt dabei dem Open-BIM-Ansatz, welcher eine neutrale, softwareunabhängige Datenbasis verfolgt. Dieses Konzept kann von allen Projektbeteiligten genutzt und mit seinen gewerkspezifischen Inhalten erweitert werden, unabhängig vom verwendeten Autorenwerkzeug. (Borrmann et al., 2021, pp. 96–98)

Das IFC-Datenschema ist objektorientiertes Datenmodell und basiert auf einer strukturierten Vererbungshierarchie. Dazu wird die konzeptionelle Schemasprache EXPRESS für die formale Spezifizierung von Klassen angewendet. Es erlaubt, sowohl geometrische als auch alphanumerische Informationen über Bauteile, Prozesse und Beziehungen in einem Bauprojekt präzise abzubilden. Spezifischer werden dabei Klassen (*Entitäten*) wie Wände, Decken oder Fenster beschrieben, die Eigenschaften (*Attribute*) sowie Beziehungen (*Relationen*) zu anderen Objekten besitzen. Über Vererbungen werden gemeinsame Merkmale in einer Hierarchie organisiert, zum Beispiel erben *IfcWall* oder *IfcColumn* allgemeine Eigenschaften von übergeordneten Klassen wie *IfcBuildingElement*. Zusätzlich erlaubt die Struktur, Objekte mit Eigenschaftensets (*Property Sets*) und Beziehungen (*Relationships*) flexibel zu erweitern, ohne die Grunddefinition zu verändern. (Borrmann et al., 2021, pp. 96–98).

Nachfolgende Erörterungen und IFC-Auszüge beziehen sich auf die Version «IFC4X3_ADD2». (buildingSMART International, 2024a)

2.5.1 Räumliche Strukturierung des Projektes nach dem IFC-Datenschema

Alle Klassen, die eine räumliche Semantik erben, erhalten Attribute und Eigenschaften von der Klasse *IfcSpatialStructureElement*. Diese gliedert sich wiederum in folgende Struktur:

- *IfcSite*: Repräsentiert des Baufeldes, auf dem das Gebäude steht.
- *IfcFacility* ist das gesamte Bauwerk.
- *IfcBuilding*: Bezeichnet das eigentliche Gebäude.
- *IfcBuildingStorey* definiert die einzelnen Geschosse innerhalb des Gebäudes.
- *IfcSpace* bezeichnet einen Raum innerhalb eines Geschosses, der eine bestimmte Funktion erfüllt wie: Büro, Flur usw.
- *IfcElement* stellt die einzelnen Bauelemente des Gebäudes dar, wie: Wände, Stützen, Decken, Fenster, Türen usw.

(Borrmann et al., 2021, p. 108).

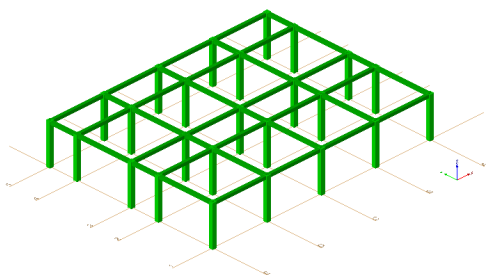


Abbildung 17; Achsen im digitalen Bauwerksmodell (BuildingSmart, 2024)

Dabei weist jedes Element eine Globally Unique Identifier (GUID) mittels *IfcGloballyUniqueId* auf. Diese codierte Zeichenfolge, dient zur eindeutigen Identifizierung eines IFC-Objektes, ist global eindeutig und ist eine automatisch generierten 128-Bit-Zahl.

Zur genaueren Strukturierung eines Gebäudegrundrisses werden, wie in der Abbildung 17 ersichtlich, Achsen in Form eines Rasters über den Grundriss gelegt. Im IFC-Datenschema können Achssysteme als eigenständige Objekte mit den Entitäten *IfcGrid* für den Hochbau oder *IfcAlignment* für

den Tiefbau abgebildet werden, wodurch sie sich auch softwaregestützt zur Navigation und Analyse nutzen lassen.

Das Raster definiert sich im flachen 3D-Raum und dient der Positionierung von Bauteilen. Die Lage des Rasters *ObjectPlacement* wird durch ein 3D-Koordinatensystem bestimmt. Dadurch kann das Raster in der Draufsicht, im Schnitt oder in einer beliebigen Ausrichtung relativ zum Weltkoordinatensystem eingesetzt werden. Die XY-Ebene des 3D-Koordinatensystems dient zur Anordnung der Rasterachsen, welche durch 2D-Kurven dargestellt werden (z.B. Linie, Kreis, Bogen, Polylinie). Die Platzierung kann sich auch auf die Objektlage anderer Produkte oder Raster beziehen so auch für die Taktplanung in Verbindung mit der Projektplanung.

2.5.2 Räumliche Objekte im IFC-Datenschema

Räumliche Elemente wie *IfcWall*, *IfcSlab*, *IfcWindow* usw. gehören in der Vererbungs-Hierarchie zur Entität *IfcElement*. Sie sind nicht nur 3D-Körper, sondern jeweils ein verbundenes Element, welches in einer Beziehung mit Informationen ist. Es steht somit in einem hierarchischen, räumlichen und semantischen Kontext. So weist jedes dieser Elemente Angaben zur Geometrie, zum räumlichen Standort, zu funktionalen Beziehungen sowie zu Material und Aufbau auf. Eine Wand weist beispielsweise Informationen zu ihren Abmessungen, zu ihrer Rasterposition, zu integrierten Bauteilen wie: Türen, zu Oberflächenmaterialien sowie zu ihrem Schichtaufbau auf. Um geometrische Darstellungen mit einer Wand *IfcWall* zu verbinden, wird dazu die Entität *IfcProductRepresentation* verwendet. Jedes physische Objekt kann mit Geometrien wie: Volumenkörper, Linien, Flächen usw. verknüpft werden.

2.5.3 Bauprozessplanung im IFC-Datenschema

Die Bauprozessplanung im IFC-Datenschema basiert auf den *IfcProcessExtension*, welche eine zentrale Erweiterung des im *IfcKernel* beschriebenen *IfcProcess* darstellt. Es ermöglicht die strukturierte Abbildung von Bauprozessen, indem Abläufe, Terminierungen, Verfahren und Ressourcen erfasst werden, die für die Ausführung der Bauarbeiten erforderlich sind. Damit werden Prozessinformationen analog zu Produktinformationen in Klassen modelliert und für den Austausch in gängigen Anwendungen nutzbar gemacht. (buildingSMART International, 2024b)

In Abbildung 18 ist ein Unified Modeling Language (UML)-Diagramm mit den Bereichen: Produkt (gelb), Prozess (gelb) und Kosten (violett) dargestellt und wie diese miteinander in Beziehung stehen. Damit eine einzelne Arbeitstätigkeit eindeutig beschrieben werden kann, dient die Entität *IfcTask* als identifizierbare Einheit im Bauwerksmodell.

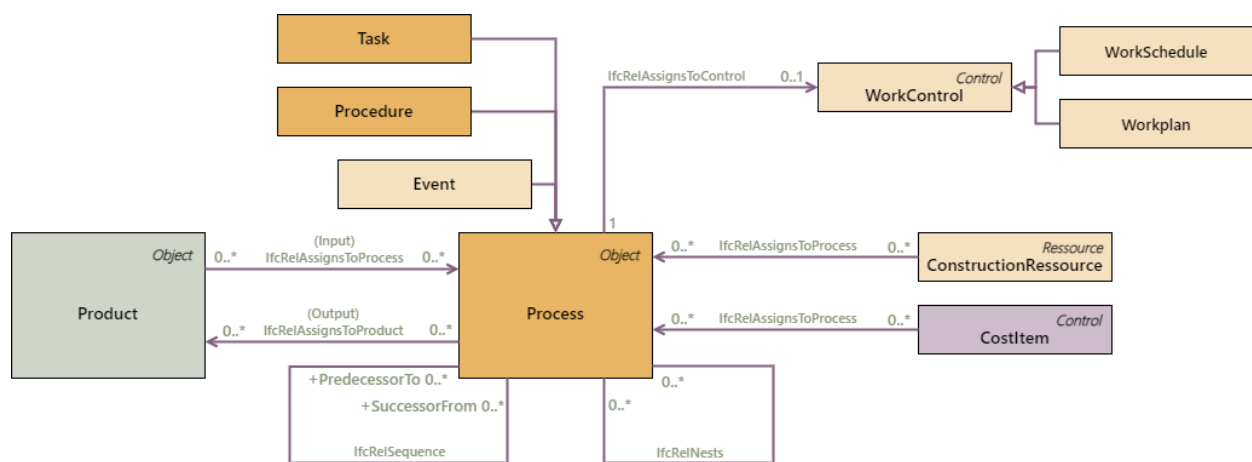


Abbildung 18; Bauprozessplanung im IFC-Datenschema abgebildet als UML-Diagramm (adaptiert von Schildknecht, 2023)

Im Zentrum steht das Objekt *Process*, das sämtliche Tätigkeiten und Abläufe auf der Baustelle beschreibt, wie beispielsweise das Betonieren einer Wand oder das Verlegen eines Bodens. Ein solcher Prozess kann unterschiedliche Formen annehmen: als *Task* (konkrete Aufgabe), *Procedure* (Vorgehensweise) oder *Event* (Ereignis, z.B. eine Prüfung, Freigabe oder eine Mängelmeldung).

Weiter kann ein *Prozess* über das Objekt *WorkControl* mit objektiver Beziehung *IfcRelAssignsToControl* in einen übergeordneten Arbeits- oder Terminplan eingebunden werden. Diese Pläne werden in Form von *WorkPlan* oder *WorkSchedule* abgebildet und ermöglichen eine strukturierte zeitliche Steuerung des Bauablaufs. Die Beziehung zwischen Prozess und Plan wird durch *IfcRelAssignsToControl* hergestellt.

Ein weiterer wichtiger Bestandteil des Modells ist die zeitliche Reihenfolge der Prozesse, die durch *IfcRelSequence* abgebildet wird. Dies ermöglicht die Festlegung von Vorgänger- und Nachfolgerbeziehungen zwischen Prozessen, um beispielsweise sicherzustellen, dass erst die Schalung errichtet wird, bevor der Beton eingebracht werden kann.

Zusätzlich lassen sich auch ressourcentechnische Aspekte wie: Arbeitskräfte, Maschinen oder Materialien über *ConstructionResource* einbinden. Selbes Vorgehen zur Anbindung an den Prozess ist auch für wirtschaftliche Kostenaspekte über *CostItem* vorgesehen.

Jedem Prozess können sowohl Eingaben als auch Ergebnisse zugewiesen werden. Diese können sich auf physische Bauelemente beziehen, die im Modell als *Product* dargestellt werden. Ein Prozess kann also entweder ein Produkt verwenden beispielsweise etwa eine Schalung als Vorbereitung oder ein Produkt erzeugen, wie zum Beispiel eine fertige Betonwand. Die Zuweisung erfolgt über die Beziehung *IfcRelAssignsToProcess*, wobei zwischen «Input» und «Output» unterschieden wird.

2.5.4 Bauprogramm im IFC-Datenschema

Für eine sinnhafte Nutzung des Bauprogramms zusammen mit dem digitalen Bauwerksmodell müssen die Strukturen des IFC-Datenschemas wie vorangehend beschrieben eingehalten werden. Ist dies gegeben, kann das zu einer guten Planungssicherheit führen, da in Echtzeit ein aktuelles Bauprogramm im digitalen Bauwerksmodell abgebildet werden kann. Wie in der Abbildung 19 ersichtlich wird die Aufgabenabfolge des Bauzeitplans anhand eines Gantt -Diagramm dargestellt.

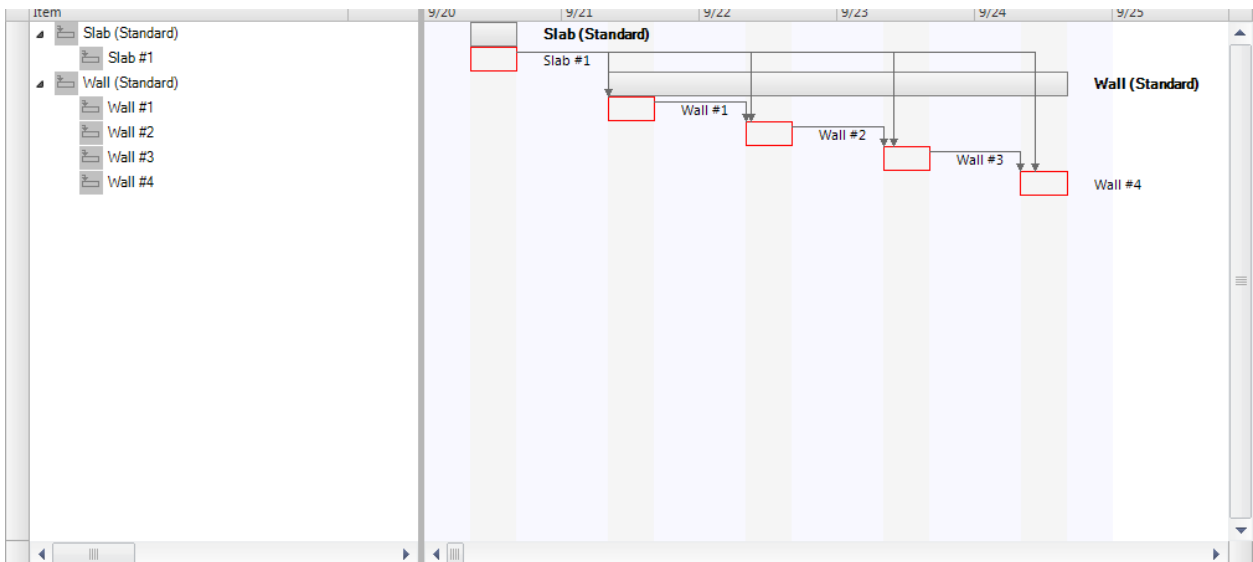


Abbildung 19; Aufgabenabfolge (Building Smart, 2024)

Am Beispiel eines exemplarischen Hochbaus einer Betondecke und vier Wänden wird im Folgenden die Verknüpfung zwischen den Bauteilen und deren Abfolge am Bau dargestellt. Den jeweiligen Bauelementen werden Aufgaben zugeordnet. Diese sind so angeordnet, dass die Wände nach Fertigstellung der Decke seriell gebaut werden. Sie sind in Summenaufgaben für die Platte und die Wände verschachtelt. Ausserdem haben die Decke und die Wände in diesem Beispiel eine lokale Platzierung, eine Körperformdarstellung und eine Materialzuweisung. (buildingSMART International, 2024c).

Daraus schliesst sich das die zeitliche Koordination von Bauteilen und Aufgaben in Projekten beispielsweise durch die Entitäten *IfcWall* in Kombination mit *IfcTaskTime* sichergestellt wird (buildingSMART International, 2024c). Dabei besitzt jede *IfcWall* über die Zuordnung zu einer *IfcTask* einen Bezug zu spezifischen Termin- und Fortschrittinformationen. Wenn ein Objekt eine bestimmte Funktionalität eines andern Objektes benötigt werden diese beiden Elemente mit einer binären Assoziation verbunden (Borrmann et al., 2021, p. 63).

Die in Tabelle 4 beschriebenen Attribute werden von IFC4X3_ADD2 für das Bauprogramm vorgesehen. (buildingSMART International, 2024d)

Tabelle 4: Vordefinierte Attribute von *IfcTask* für das Bauprogramm

Eigenschaften (Attribute)	Beschrieb
Duration Type	Arbeitszeit oder verstrichene Zeit.
ScheduleDuration	<i>IfcDuration</i> : Dieser Wert gibt die geplante Gesamtdauer der Aufgabe an, also der Differenz zwischen <i>ScheduleFinish</i> und <i>ScheduleStart</i> .
ScheduleStart, ScheduleFinish	Diese beiden Datumswerte geben den geplanten Beginn bzw. das geplante Ende der Aufgabe an. Dabei gibt <i>ScheduleStart</i> beispielsweise das geplante Startdatum an, während <i>ScheduleFinish</i> das voraussichtliche Fertigstellungsdatum angibt.
EarlyStart, Early Finish	Das früheste Datum, an dem eine Aufgabe begonnen/abgeschlossen werden kann.
LateStart, LateFinish	Das späteste Datum, an dem eine Aufgabe begonnen/abgeschlossen werden kann
FreeFloat	Der Zeitraum, in dem der Beginn oder das Ende einer Aufgabe ohne Auswirkungen auf den Gesamtarbeitsplan verschoben werden kann.
TotalFloat	Die Differenz zwischen der für die Ausführung einer Aufgabe verfügbaren Zeit und der geplanten Dauer der Aufgabe.
IsCritical	Eine Wertung, welche angibt, ob eine geplante Aufgabe ein kritischer Punkt innerhalb des Programms ist.
ActualDuration	<i>IfcDuration</i> : Dieser rechnerische Wert ergibt sich aus <i>ActualFinish</i> minus <i>ActualStart</i> und stellt die real benötigte Zeit für die Ausführung dar. Abweichungen im Vergleich zu <i>ScheduleDuration</i> zeigen andererseits Verzögerungen oder Effizienzsteigerungen an.
ActualStart, ActualFinish	Diese Felder dokumentieren die tatsächlich gemessenen Anfangs- und Endzeitpunkte der Aufgabe und sind entscheidend für Soll-Ist-Vergleiche im Bauablauf.
RemainingTime	Die verbleibende Zeit bis zur Fertigstellung einer Aufgabe. Es handelt sich um einen prognostizierten Wert. Der Wert wird entweder als verstrichene Zeit oder als Arbeitszeit angegeben, die durch <i>DurationType</i> definiert ist.
Completion	<i>IfcPositiveRatioMeasure</i> : Grad der Fertigstellung und aktueller Bearbeitungsfortschritt der Aufgabe als Verhältnisswert (0,0–1,0) mit der Wand. Ein Fertigstellungs-Wert von 0,33 bedeutet, dass 33 % der Wandarbeiten abgeschlossen sind.

2.5.5 Datenaufbereitung vom IFC-Datenschema hin zur Weiterverarbeitung mittels Low-Code

Die Software-Umgebung von Microsoft Power Platform kann IFC-Dateien nicht direkt lesen (buildagil, 2025). Es ist deshalb ein Zwischenschritt über eine Comma Separated Values (CSV)-Datei nötig. Dazu können mit einer Extract, Transform, Load (ETL)-Schnittstelle die notwendigen Bauteilinformationen strukturiert aus einer IFC-Datei extrahiert und in eine tabellarische CSV-Datei überführt werden. Die Übersetzung kann dabei mit einem Python-Script beispielsweise in der Entwicklungsumgebung Visual Studio Code erfolgen. Die Schnittstelle dient somit als Brücke zwischen dem semantisch und geometrisch komplexen IFC-Modell und einer einfach weiter verarbeitbaren Tabellenstruktur.

Als Datenquelle dient eine IFC-Datei, welche ein digitales Bauwerksmodell mit unterschiedlichen Entitäten, beispielsweise *IfcSpace* oder *IfcWall*, beinhaltet. Die Daten können anschliessend mithilfe der Python-Bibliothek *ifcopenshell* extrahiert, die speziell für das Parsen und Interpretieren von IFC-Modellen entwickelt wurde. Zusätzlich kann mit dem Einsatz von *ifcopenshell.geom* der Zugriff auf die geometrischen Eigenschaften der Bauteile ermöglicht werden. (*IfcOpenShell Contributors*, 2025)

Im Rahmen der Transformation können relevante Informationen wie: die GUID, der Typ des Bauteils, das zugehörige Geschoss, der Raum Name sowie geometrische Kenngrössen wie: Länge, Höhe und Dicke verarbeitet. Die Werte können dabei primär aus den PropertySets der IFC-Datei gelesen beziehungsweise extrahiert werden. Das gleiche Vorgehen kann auch in umgekehrter Richtung angewendet werden, um die neu angereicherten Bauteilinformationen wieder in die IFC-Datei zurückzuschreiben.

Den Bauwerksmodelldaten können zusätzlich weitere Spalten zum Hinzufügen von weiteren Attributen gemäss dem IFC-Datenschema für die Abbildung vom Bauprozess angefügt werden. Diese Felder dienen der späteren Anreicherung mit Planungs- und Statusinformationen, sind im aktuellen Export jedoch noch leer. Anschliessend

kann der gesamte Datenbestand in eine CSV-Datei mit definierter Spaltenstruktur und Semikolon als Trennzeichen geschrieben werden. (buildingSMART International, 2024d)

Anschliessend wird die CSV-Datei manuell zur Tabelle formatiert und als Excel-Datei gespeichert. Somit wird die Excel-Datei zur Datenbank der relevanten Bauteilinformationen. Excel als leistungsstarkes Werkzeug für die Datenverarbeitung, besonders für Nicht-Programmierer und könnte so auch direkt in die bestehenden Geschäftsprozesse eingebunden werden. (Chuong Mep, 2024)

In einem letzten Schritt wird die externe Excel-Datei mit Microsoft Power Apps verbunden. Dafür ist es erforderlich, die Datei zunächst in einem Cloud-Speicherdienst wie OneDrive for Business abzulegen. Der Zugriff auf die Daten erfolgt anschliessend über den in Microsoft Power Apps integrierten Excel-Connector. (Microsoft, 2025g)

2.6 Erwähnungen von exemplarischen Use-Cases in der Literatur

Die wissenschaftliche Literatur zu Low-Code-Plattformen im Kontext der Baubranche ist derzeit noch sehr begrenzt vorhanden. Während Low-Code in Branchen wie der Fertigung, Logistik oder dem Finanzwesen bereits intensiv diskutiert wird, beispielsweise zur Automatisierung von Workflows oder zur schnellen Prototypentwicklung, fehlen im Bauwesen bislang systematische Untersuchungen. Dies deutet darauf hin, dass die Potenziale dieser Technologie im Bauwesen noch weitgehend ungenutzt sind. Entscheidend sei dabei, unabhängig der Branche, aber die ganzheitliche Betrachtung des Prozesses, um eine sinnvolle Digitalisierung zu bewirken.

Mit Blick auf das letzte Jahrzehnt gab es viele disruptive technologische Innovationen, die vor allem durch die digitale Transformation eingeführt wurden. Dabei ist zu beobachten, dass neue Geschäftsmodelle wie beispielsweise Uber und Airbnb die Märkte grundlegend veränderten. Diese Unternehmen haben ihr gesamtes Geschäftsmodell auf cloudbasierten Technologieplattformen aufgebaut. Sie ermöglichen es ihnen, ihre Aktivitäten schneller und einfacher zu skalieren als dies mit traditionellen Technologiearchitekturen möglich wäre. (Metrolho et al., 2020, p. 8601)

Die digitale Transformation findet in diesen Firmen jeweils auf unterschiedlichen Unternehmensebenen statt. Metrolho et al. beschreiben zusammenfassend, dass sie sich unternehmensintern in drei Stadien beobachten lässt:

1. In der ersten Phase werden, die heute in Papierform vorliegenden Dokumente in ein digitales Informationssystem überführt.
2. In der zweiten Phase werden die bestehende digitale Prozessautomatisierung und auch die manuelle Prozessautomatisierung, meist durch Integration von Systemen/Anwendungen, ersetzt.
3. In der dritten Phase werden die Geschäftsmodelle, in denen die Unternehmen tätig sind, vollständig neugestaltet. (Metrolho et al., 2020, p. 8601)

Apps sind dann sinnvoll, wenn Informationen auf die mobilen Endgeräte als auch auf den stationären Rechnern verfügbar sind. Dabei sind der Bedienungskomfort und die Benutzerfreundlichkeit wichtiger als die Funktionsfähigkeit im Offline-Modus. Solche Apps sind dabei grösstenteils für Controlling- oder Management-Informationssysteme angedacht, um jederzeit und an jedem Ort auf zentrale Unternehmensdaten zugreifen zu können. Der grösste Nutzen entsteht dann, wenn die Zusammenstellung der App direkt auf dem mobilen Endgerät durch die Endnutzerinnen und Endnutzer selbst erfolgen kann. (Noack & Fedtke, 2025, p. 45)

Die am weitesten verbreitete Verwendung von Low-Code sei in mittleren bis grösseren KMU anzutreffen. Dort werden sie gerne für die Anbindung der Standardsoftwares gebraucht. Häufig wird diese Erweiterung bereits von einigen CRM- und ERP-Systemen angeboten um Standardchnittstellen selbständig zu erweitern. (Noack & Fedtke, 2025, p. 44)

2.6.1 Automatisierung vom Baustelleninventar mittels Low-Code

Das Projekt betrifft den Bau eines grossen Tunnels in der Schweiz. Die betrieblichen Herausforderungen ergeben sich aus einer typischen Situation einer manuellen Datenverarbeitung auf Baustellen. Für jeden Betonlieferwagen erhält das Bauteam vor Ort einen entsprechenden Lieferschein. Die relevanten Informationen des Dokuments (z.B. Eingangsnummer, Materialart, Datum, Menge usw.) werden manuell in eine Tabellenkalkulation übertragen, um den Überblick über die gelieferten Materialien zu behalten. Die Informationen werden dann weiterverarbeitet, um zusätzlich ein Dashboard zur Visualisierung der Informationen zu haben. Dieses Verfahren birgt einige Herausforderungen:

Erstens ist die manuelle Übertragung der Informationen vom Lieferschein in die Tabellenkalkulation zeitaufwändig und fehleranfällig. Die Mitarbeiter wenden täglich Zeit für diese unproduktive Aufgabe auf, anstatt sich auf die

Wertschöpfung des Projekts zu konzentrieren.

Zweitens führt die zuständige Person diese Aufgabe in vielen Fällen nicht regelmässig aus. Dadurch entsteht ein Stau an Lieferscheinen, die nicht bearbeitet werden, so dass das Projektteam nicht über genaue und rechtzeitige Informationen über die Menge des vor Ort eingegangenen Materials verfügt. Obwohl dies nicht im Rahmen der Prozessverbesserung berücksichtigt wird, wirkt sich die Verzögerung bei der Informationsverarbeitung auch auf nachgelagerte Arbeiten aus. Da es keinen Überblick über das eingegangene Material gibt, ist das Finanzteam nicht in der Lage, Rechnungen zentral zu bearbeiten, was zu Verzögerungen bei den Zahlungslieferanten führt.

Das Team wollte einige der Ineffizienzen im Prozess durch den Einsatz der Low-Code Microsoft Power Platform beheben, was teilweise erfolgreich war. Gleichzeitig traten jedoch an anderen Stellen im Geschäftsprozess unerwartete prozessuale Störungen auf, welche aber auf die Low-Code-Technologie zurückzuführen sind. (Martinez & Cisterna, 2023, p. 201)

2.6.2 Reservierung von Elektro-Ladestationen mittels Low-Code-Anwendung

Für die Fallstudie wurde ein Buchungssystem für Ladestationen von Elektrofahrzeugen entwickelt, da es zuvor keinen koordinierten Zugang gab. Bisher nutzten die Mitarbeiter die Ladestationen nach dem Prinzip «Wer zuerst kommt, mahlt zuerst», was zu Blockaden und mangelnder Transparenz führte. Mit dem erwarteten Anstieg der Anzahl von E-Autos erwartete das Unternehmen eine Verschärfung dieses Problems und strebte eine bessere Ausnutzung der bestehenden Anlagen an.

Die Lösung besteht in einem digitalen Buchungssystem auf Basis von INTREXX, das über das Intranet zugänglich ist. Mitarbeiter können Zeitfenster reservieren und erhalten kurz vor Ablauf des Zeitfensters eine E-Mail-Benachrichtigung. Weiter werden Firmenwagen priorisiert behandelt. Das System wird als Client-Server-Anwendung umgesetzt. Während der Client die Reservierungen anzeigt, verwaltet der Server Buchungen und versendet Benachrichtigungen über Microsoft Exchange.

Die Erkenntnisse aus der Fallstudie sind, dass Citizen Developer in kurzer Zeit für die Entwicklung von Low-Code-Anwendungen geschult werden können, während die Verwaltung der Anwendung weiterhin von IT-Experten übernommen wird. Für einige Aufgaben sind Programmierkenntnisse erforderlich etwa bei der Definition von Datenstrukturen oder bei der Modellierung ereignisgesteuerter Anwendungen. Auch für die Erstkonfiguration von Schnittstellen zu Drittsystemen ist IT-Know-how erforderlich. Eine weitere erwähnenswerte Erkenntnis ist, dass Citizen Developer dazu neigen, die Entwicklungsaufgabe am Frontend anzugehen, während professionelle Entwickler eher am Backend beginnen. (Weidmann et al., 2024, p. 12)

2.6.3 Bauteilprüfung vor Ort mit einer Low-Code-Anwendung

Überprüfungen und Abnahmen vor Ort können viel Zeit in Anspruch nehmen. In der bisherigen Arbeitsweise wurden Papierformulare genutzt um den Servicegrad, die Eignung und Qualität der Materialien sowie die Arbeitsqualität zu messen. Die App ist für die Techniker vor Ort, um diese Abnahmen mithilfe einer Applikation zeitsparender und mit mehr Genauigkeit zu arbeiten. Weiter sind die Daten nun für alle in der Unternehmung zugänglich. (AppSheet, 2019, p. 8)

Alternativ bietet die Autodesk Construction Cloud in ihrem Autodesk Build-Modul ein Formularwerkzeug an. Mit solchen Formularen lassen sich digitale Arbeitsprozesse wie Zwischenabnahmen deutlich effizienter gestalten. Intelligente PDF-Dateien können hochgeladen und in Formularvorlagen überführt werden, wodurch eine strukturierte Datenerfassung ermöglicht wird. Dabei lassen sich automatische Felder definieren, die projektbezogene Informationen selbständig einfügen. Dies reduziert sowohl den zeitlichen Aufwand als auch die Fehleranfälligkeit bei der Eingabe. Zudem besteht Flexibilität bei der Gestaltung der Vorlage: Es kann festgelegt werden, ob zusätzliche Informationen wie Wetterdaten oder Anmerkungsfelder integriert werden sollen. Ein weiterer Vorteil zeigt sich in der kollaborativen Nutzung. Rollen wie Bearbeiter, Prüfer oder Betrachter können gezielt zugewiesen werden, was eine transparente und nachvollziehbare Zusammenarbeit unterstützt. (Autodesk Inc., 2025a)

2.7 Perspektiven durch die Anwendung von Low-Code

Neue Technologien wecken oft Begeisterung und die Hoffnungen für Verbesserung. Leider gibt es auch immer die Gegenseite, welche Herausforderungen mit sich bringt. Auch Low-Code wird durch Berichte über erfolgreiche Anwendungen von Low-Code-Lösungen in der Industrie beflügelt und durch optimistische Einschätzungen des zukünftigen wirtschaftlichen Potenzials des Low-Code-Sektors weiter genährt (Shah, 2020).

Leider wurde bei vielen bisherigen Arbeiten zur Low-Code-Entwicklung eine relativ unkritische Haltung eingenommen. Konkret wurden viel zu wenige Anstrengungen unternommen, um ein klares und eindeutiges Konzept für diese Systeme zu entwickeln. (Bock & Frank, 2021, p. 734).

So hat Waszkowski die Low-Code-Plattformen in seiner Publikation wie folgt beschrieben:

„Die Low-Code-Plattform ist eine Reihe von Tools für Programmierer und Nicht-Programmierer. Sie ermöglicht die schnelle Erstellung und Bereitstellung von Geschäftsanwendungen mit minimalem Aufwand für das Schreiben in einer Programmiersprache und erfordert den geringstmöglichen Aufwand für die Installation und Konfiguration von Umgebungen sowie für Schulung und Implementierung“ (Waszkowski, 2019, p. 376)

Auch Chang & Ko erwähnen den spürbar positive Unterton, welcher oft bei neueren Technologien zu finden ist (Chang & Ko, 2017). Es wird vermutet, dass der Fokus aber darauf liegt, ein Projekt rechtzeitig und innerhalb der kalkulierten Kosten abzuschliessen, anstatt es umfassend zu integrieren. Was wiederum die Grundlage ist, um in der Branche erfolgreich konkurrieren zu können. Daher müssen die Vorteile also schnell sichtbar sein (Linderoth et al., 2018). Dies aber nicht nur auf der operativen Ebene: die «Machermentalität» wird in der Führungsebene von Bauunternehmen als tragendes Identitätsbildendes Element betrachtet (Löwstedt & Räisänen, 2014).

Dies bedeutet, dass der Kunde als einer der mächtigsten Akteure angesehen werden kann. Da Kunden vor allem den niedrigsten Preis bevorzugen, sehen sich die Dienstleister gezwungen ihre Unternehmensausrichtung auch darauf auszurichten. Auch kurzfristige und marktorientierte Beziehungen formen die Angebote der Dienstleister. Linderoth et al. beschrieben, dass somit der Kunde und das Produkt die zentralen Treiber sind, um die Entwicklung der Merkmale in der Bau- und Konstruktionsbranche im Laufe der Zeit zu verstehen. Sein Fokus liegt auf den Investitionskosten, wobei er sich der Lebensdauer der Produkte bewusst ist. Ausserdem weiss er, dass der Grossteil der Kosten mit dem Betrieb und der Wartung verbunden ist. Nach den Interpretationen von Linderoth et al. ist der Fokus des Kunden also etwas paradox. (Linderoth et al., 2018).

Sanchis et al. beschreiben in ihrer Publikation, dass Low-Code-Entwicklungsplattformen durch interne, nicht ausgelagerte Entwicklung die Datensicherheit erhöhen und mittels visueller Konfiguration vorgefertigter Codebausteine die Entwicklungszeiten um das Fünf- bis Zehnfache verkürzen. Dies führt zugleich zu einer signifikanten Kostensenkung. Sie heben ausserdem hervor, dass die Verringerung der Komplexität und des Wartungsaufwands durch geringen Eigencode sowie die intuitive Einbindung von Fachanwendern als Entwickler eine präzisere Abbildung der Geschäftsanforderungen ermöglicht. Zudem fördern Low-Code-Plattformen die Minimierung instabiler oder inkonsistenter Anforderungen, indem sie die rasche Erstellung minimal funktionsfähiger Prototypen ermöglichen und so eine Verschwendung von Ressourcen für nicht wertschöpfende Features verhindern. Eine Umfrage, dargestellt in Abbildung 20, wurde unter mehr als 3'300 IT-Fachleuten durchgeführt. Sie zeigt, dass 66 % der Befragten Low-Code-Plattformen vor allem einsetzen, um die digitale Transformation zu beschleunigen und die Reaktionsfähigkeit von Unternehmen zu erhöhen. Zudem nennen 45 % die Verringerung der Abhängigkeit von schwer verfügbaren technischen Fachkräften als zentrales Motiv (Sanchis et al., 2019, p. 3).

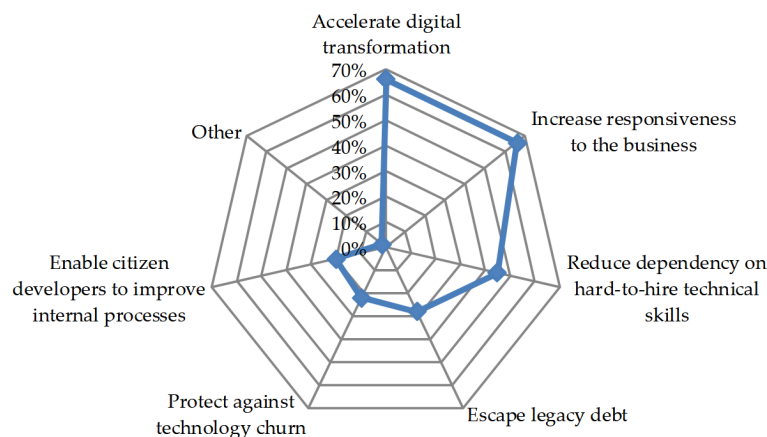


Abbildung 20; Hauptgründe, um Low-Code development Plattformen zu nutzen (Sanchis et al., 2019, p. 3)

Neben den bereits genannten Aspekten weisen weitere Quellen auf zusätzliche Chancen und Herausforderungen hin, die als Begründung für den Einsatz von Low-Code-Technologien angeführt werden. Im Folgenden werden diese thematischen Schwerpunkte näher dargestellt.

2.7.1 Chancen durch den Einsatz von Low-Code-Technologien

Steigerung der Produktivität

Unternehmen, die auf Low-Code-Anwendungsplattformen setzen, profitieren von einer signifikanten Steigerung ihrer Produktivität. Daraus ergibt sich die Chance, zentrale, abteilungsbezogene und nutzerbezogene IT-Funktionen durch multifunktionale Werkzeuge zu stärken (Vincent et al., 2020, p. 1). Die Produktivitätssteigerung eröffnet die Möglichkeit, Anwendungsentwicklungen nicht nur zu beschleunigen, sondern zugleich Ressourcen effizienter zu nutzen und damit Projektbudgets zu entlasten (Bock & Frank, 2021, p. 733). Eine wesentliche Chance liegt in der Reduzierung der Entwicklungszeit, da wieder verwendbare Module und visuelle Konfigurationsmöglichkeiten den manuellen Programmieraufwand erheblich verringern können. Dadurch lassen sich Innovationszyklen verkürzen und gleichzeitig die Qualität der Softwarelösungen erhöhen (Alsaadi et al., 2021, p. 123).

Verkürzung der manuellen Bearbeitungszeit

Es hat sich gezeigt, dass sich innerhalb kurzer Zeit eine Low-Code-Lösung entwickeln lässt, die die meisten manuellen Verarbeitungsschritte eliminiert und somit die Bearbeitungszeit verkürzt, während gleichzeitig ein zufriedenstellender Überblick über die Optimierung erhalten bleibt (Vikebø & Sydvoid, 2019, p. 2). Daraus ergibt sich die Möglichkeit, interne Durchlaufzeiten zu reduzieren und zugleich die Kosten für die Entwicklung und Wartung von Unternehmenssoftware zu senken. Die verkürzte Bearbeitungszeit schafft zudem die Chance, die Agilität von Projektteams zu fördern, indem Ressourcen frei werden, die ansonsten in langwierigen Programmier- und Testzyklen gebunden wären (Bock & Frank, 2021, p. 733).

Wahrung der Unternehmensgeheimnisse

Unternehmen, die ihren Mitarbeitenden die Entwicklung von Anwendungen ohne tiefgreifende technische Fachkenntnisse ermöglichen, schaffen damit die Chance, sensible Aufgaben bewusst innerhalb der eigenen Organisation zu verankern. Daraus ergibt sich die Möglichkeit, die Vertraulichkeit und den Schutz unternehmensinterner Daten zu stärken (Sanchis et al., 2019, p. 2). Da keine externen Dienstleister involviert sind, bleiben Einblicke in betriebliche Prozesse und Geschäftslogiken exklusiv innerhalb der Belegschaft und unterliegen so einem höheren Mass an Kontrolle. Ebenso entsteht die Chance, durch die direkte Einbindung der Endanwender deren Rolle zu stärken, da sie ihre Bedürfnisse unmittelbar in den Entwicklungsprozess einbringen und passgenaue Lösungen gestalten können. So ergibt sich ein innovationsfreundliches Umfeld, in dem Datenschutz und Empowerment als Chance Hand in Hand gehen (Bock & Frank, 2021, p. 733).

Schnelles Einwirken auf Veränderungen

Durch die Vorfertigung des Grossteils des Codes eröffnet Low-Code die Chance, Anwendungen durch visuelle Konfiguration effizient zu erstellen und anzupassen. Damit sinkt der Entwicklungsaufwand erheblich, und Applikationen können in kürzester Zeit einsatzbereit sein (outsystems, 2019). Eine Forrester-Studie belegt eine Beschleunigung des Entwicklungsprozesses um das Fünf- bis Zehnfache, wodurch zusätzlich Einsparpotenziale entstehen (Richardson & Rymer, 2016). Die verkürzten Entwicklungszyklen bieten Organisationen die Möglichkeit, ihre Software rasch an veränderte Anforderungen anzupassen, wodurch Agilität und Reaktionsfähigkeit gestärkt werden. Zudem lassen sich Anwendungen plattformübergreifend entwickeln und Erweiterungen innert kürzester Zeit bereitstellen, was die Wettbewerbsfähigkeit von Unternehmen nachhaltig fördert (Metrolho et al., 2020, p. 8602). Bei Low-Code-Lösungen sind die Entwicklungskosten sehr niedrig und somit am attraktivsten für kleine Anwendungen. (Moskal, 2021, p. 56)

Reduzierung der Komplexität durch den Einsatz von Fachexpertinnen und -experten

Low-Code-Entwicklungsplattformen eröffnen die Chance, den Prozess der Softwareerstellung wesentlich zu vereinfachen, indem Anwendungen aus vorgefertigten Bausteinen zusammengesetzt werden. Aufgrund des geringen Umfangs an zu pflegendem Code entfallen aufwendige Wartungsarbeiten nahezu vollständig, wodurch eine permanente Übereinstimmung zwischen angebotenen Services und sich wandelnden Geschäftsanforderungen sichergestellt wird (outsystems, 2019).

Da keine spezifischen Programmierkenntnisse oder jahrelange Erfahrung erforderlich sind, können auch

Fachexpertinnen und -experten ihre Domänenkompetenz direkt in die Anwendung einbringen und dabei vertraute Modellierungsformalismen auf der passenden Abstraktionsebene nutzen (Metrolho et al., 2020, p. 8602). Low-Code-Plattformen zeichnen sich dadurch aus, dass sie eine einfache und intuitive Benutzeroberfläche als einheitliche Entwicklungsumgebung bereitstellen. Dadurch können auch Fachexperten ohne technologische Vorkenntnisse zu App-Entwicklern werden und ihre tiefgehenden Geschäftskennnisse direkt in den Entwurf einbringen (Waszkowski, 2019, pp. 376–381). Tatsächlich arbeiten 44 % der Low-Code-Nutzer eng mit der IT-Abteilung zusammen, um Geschäftsanwendungen passgenau umzusetzen (outsystems, 2019).

Durch die Fähigkeit, schnell minimal funktionsfähige Prototypen zu erzeugen und so Anforderungen unmittelbar zu validieren, werden inkonsistente oder instabile Anforderungen minimiert und teure Fehlentwicklungen vermieden (Richardson & Rymer, 2016). Im Gegensatz zur klassischen Softwareentwicklung, für die eine Vielzahl getrennter Werkzeuge erforderlich ist, vereinen Low-Code-Plattformen alle notwendigen Tools von Modellierungsumgebungen über DBMS-Anbindungen bis hin zu GUI-Editoren in einer integrierten Umgebung. Dadurch werden Wechselkosten und Inkonsistenzen der Implementierungsartefakte stark reduziert (Bock & Frank, 2021, p. 738).

2.7.2 Herausforderungen durch die Low-Code-Technologieanwendung

In einem Literaturüberblick über 282 wissenschaftliche Veröffentlichungen zur digitalen Transformation wurden zwei Arten von Veränderungen des Wertschöpfungsprozesses als Hindernissen für die digitale Transformation identifiziert. Einerseits bezieht sich die Trägheit auf die in einer Organisation vorhandenen Ressourcen und Fähigkeiten, die Hindernisse für Veränderungen schaffen können, die die bestehenden Prozesse in der Organisation stören könnten. Andererseits stellt der Widerstand von Mitarbeitenden ein Hindernis für die digitale Transformation dar, wenn disruptive Technologien in einer Organisation eingeführt werden. Dieser Widerstand wirft Fragen nach der geeigneten Vorgehensweise und dem Tempo der Implementierung auf und wird in der Fachliteratur häufig unter dem Begriff der Innovationsmüdigkeit diskutiert. (Vial, 2019).

Tisi et al. beschreiben in ihrer Publikation, dass Low-Code-Entwicklungsplattformen durch drei Hauptbeschränkungen gekennzeichnet sind: mangelnde Skalierbarkeit bei gross angelegten oder unternehmenskritischen Anwendungen, eine Paradigmenfragmentierung infolge herstellereinspezifischer Programmiermodelle und die Fokussierung auf rein softwarebasierte Systeme. Letztere adressieren die fachliche Abstraktion von Experten anderer Ingenieurdisziplinen nicht angemessen. (Tisi et al., 2019, p. 74).

Wie in Abbildung 21 ersichtlich ist, verzichten Unternehmen häufig auf den Einsatz von Low-Code-Plattformen, weil das Wissen über deren Funktionsweise und Potenziale fehlt. Zudem bestehen Bedenken hinsichtlich Vendor-Lock-In, eingeschränkter Flexibilität, begrenzter Skalierbarkeit sowie der Sicherheit. (Sanchez et al., 2019, p. 4)

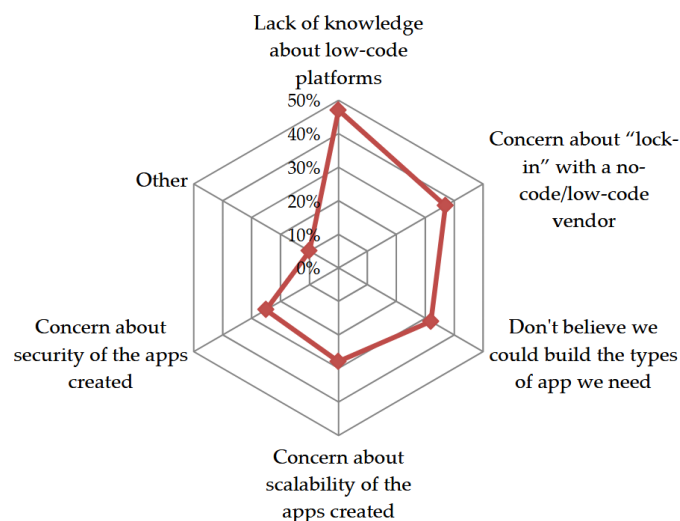


Abbildung 21; Gründe weshalb Low-Code nicht verwendet wird. (Sanchez et al., 2019, p. 3)

Herausforderungen bei der Skalierbarkeit

Kleine und mittelgroße Unternehmen könnten von der effektiven und effizienten Softwareentwicklung profitieren, die durch Low-Code-Development-Plattformen ermöglicht wird (Weidmann et al., 2024, p. 8). Denn gerade Low-Code-Plattformen stossen bei der Skalierbarkeit deutliche an ihre Grenzen, da sie primär für die Entwicklung

kleinerer Apps konzipiert sind und ihre Eignung für Grossprojekte oder unternehmenskritische Anwendungen bislang unzureichend belegt ist (Sanchis et al., 2019, p. 2). Hinzu kommt die Schwierigkeit, den notwendigen Know-how-Transfer innerhalb grosser Organisationen sicherzustellen. Ohne diesen Austausch bleibt die Nutzung solcher Automatisierungen häufig auf einzelne Fachbereiche beschränkt, sodass viele Lösungen isoliert bestehen und nicht konzernweit Anwendung finden (Alsaadi et al., 2021).

Langfristige Datendurchgängigkeit

Bei Low-Code-Plattformen stellt es eine Herausforderung dar, dass die Nutzenden keinen Einfluss auf die unterstützten Schnittstellen haben. Dabei droht die Herausforderung, dass Lock-in-Effekte entstehen, weil die in einer Plattform erstellten Schnittstellendefinitionen nur selten in andere Umgebungen übertragen werden können (Bock & Frank, 2021, p. 738).

Datenqualität und Lebensdauer der Daten, Rollen

Im digitalen Lebenszyklus von Baudaten zeigt sich eine erhebliche Diskrepanz zwischen den langfristigen Anforderungen des Facility Managements, das Informationen über Jahrzehnte hinweg vorhalten muss, und der kurzzyklischen Perspektive der IT. Die Herausforderung besteht darin, dass dadurch nicht nur die Zugänglichkeit von Daten nach zehn Jahren unsicher wird, sondern auch ihr dauerhafter Pflegeaufwand zu bewältigen ist (Linderoth et al., 2018).

Aneignen der Fertigkeiten und Hilfestellung

Die zunehmende Verbreitung von Low-Code-Entwicklungsplattformen erfordert nicht nur geeignete Werkzeuge, sondern auch gezielte Hilfestellung und Schulungen, damit Anwender ihre Modellierungsfertigkeiten effektiv ausbauen können. Die Studie von Mosquera et al. zeigt, dass zwar 58 Forschungsarbeiten die Unterstützung von Nutzern in modellbasierten Softwareentwicklungstools thematisieren, entsprechende Anleitungen in vielen kommerziellen Tools jedoch fehlen (Mosquera et al., 2024, p. 16).

Fehlende Rahmenwerke und mangelnder Support werden hauptsächlich von Citizen Developern als zentrales Hindernis wahrgenommen, da sie zwar technisch versiert sind, aber nicht das Wissen von klassischen Softwareentwickelnden haben (Khorram et al., 2020; Overeem et al., 2021).

Besonders kleine und mittelgrosse Unternehmen stehen vor der Herausforderung, eine geeignete Low-Code-Development-Plattform auszuwählen, diese in die bestehende IT-Landschaft zu integrieren und einen passgenauen Entwicklungsprozess zu etablieren. Die Bewältigung dieser Herausforderung erfordert zusätzlich eine gezielte Schulung des Personals sowie eine kontinuierliche Begleitung. (Weidmann et al., 2024, p. 8). Gleichzeitig eröffnen KI-basierte Assistenzsysteme wie Large Language Modelle (LLM) und Generative Pretrained Transformer (GPT) neue Perspektiven, um Nutzer bei Modellierungs- und Coding-Aufgaben zu unterstützen und so die Lernkurve nachhaltig zu verkürzen (Mosquera et al., 2024, p. 16).

Bei einem personellen Wechsel stellt sich immer die Frage der Nachvollziehbarkeit. Low-Code-Konfigurationen versprechen, dass diese die erstellten Codes automatisch dokumentieren und damit längerfristig verständlich bleiben. (Noack & Fedtke, 2025, p. 46)

Einfluss von Angst und Unwissen

Die Zurückhaltung gegenüber Low-Code-Technologien gründet oft auf Unsicherheit und fehlendem Wissen über die Plattformen. Viele entscheiden sich gar nicht dafür, da sie glauben, ihnen fehle das notwendige Know-how. (Sanchis et al., 2019, p. 4)

Auch wenn Low-Code die Entwicklung und Bereitstellung vereinfacht, führen Angst und Unwissen oft zu Unsicherheit im Umgang mit Projekten. Deshalb bleibt es notwendig, diese wie traditionelle Software-Implementierungen zu organisieren und klare Strukturen zu schaffen. Nur so lassen sich Qualität, Sicherheit und Kosten langfristig beherrschen. (Vikebø & Sydvoid, 2019, p. 15)

2.8 Konklusion vom Stand der Praxis, Technik und der Forschung

In Anbetracht des voran beschriebenen Stands der Praxis, Technik und Forschung lassen sich folgende Vor- und Nachteile sowie faszinierende Konzepte ableiten:

Der Bauleiter ist verpflichtet ein Protokoll also beispielsweise ein Baujournal zu führen (SIA 102, 2020, p. 29). Aus diesem lassen sich Pendenzenlisten ableiten, aber auch wesentliche Eckdaten festgehalten (SIA 102, 2020). Dies

ist einerseits die Beschreibung und Ort des Bauvorhabens, Datum, Uhrzeit sowie Witterungsbedingungen wie Wetter, Temperatur und Luftfeuchtigkeit des Aufzeichnungstags. Weiterhin gehören dazu die anwesenden Arbeitsgattungen mit der Zahl der eingesetzten Arbeiter sowie der Baufortschritt (Menz, 2020).

Um den Baufortschritt greifbar zu machen, wird die Einheit in Prozent genutzt (Makler- und Bauträgerverordnung - MaBV, 2023). Im IFC-Datenschema wird der Grad der Fertigstellung mit dem Attribut *Completion* ausgedrückt (buildingSMART International, 2024d). Für die zeitliche Koordination mithilfe des IFC-Modells werden Arbeitsgänge mit der Entität *IfcTaskTime* über die Entität *IfcRelAssignsToProcess* mit einem Produkt wie beispielsweise *IfcWall* in Beziehung gesetzt. (buildingSMART International, 2024c)

Um diese Einheit mit der Bauplanung in Relation setzen zu können wird für die zeitliche Planung und Nachverfolgung ein Gantt- oder Balkendiagramm erstellt. Auf der vertikalen Achse sind die Aufgaben und Aktivitäten aufgeführt, auf der horizontalen Achse ist der Zeitverlauf in Tagen, Wochen oder Monaten dargestellt. Die Balken markieren jeweils den Start- und Endzeitpunkt der Tätigkeit und veranschaulichen so die Dauer.

Eine kollaborative Bauprojektmanagements-Methode ist das Last Planner System. Es bezieht die ausführenden Personen direkt in den Planungsprozess ein und berücksichtigt nur jene Aufgaben, die als realistisch umsetzbar gelten. Dadurch werden Störungen minimiert, die Termintreue steigt und ein kontinuierlicher Verbesserungsprozess entsteht.

Diese Planung kann erst mit einem vertieften Verständnis der jeweiligen Arbeitsgattung und der Ganzheitlichen Betrachtung sinnhaft kalkuliert werden. Und stellt die Ausgangslage für die Nutzung des 4D-Modells dar. Wird dies detailliert nach Bauabschnitten gegliedert, können Arbeitsgattungen in ihrer sequenziellen Abfolge nachvollziehbar dargestellt werden. (Borrmann et al., 2021, p. 400)

Im Abgleich zwischen Bauzeit-Soll zu Bauzeit-Ist wird im IFC-Datenschema das Attribut (*RemainingTime*) betrachtet. Er wird berechnet aus der geplanten Dauer (*ScheduleDuration*) und der effektiv gebrauchten Dauer (*ActualDuration*). Die effektiv gebrauchte Dauer wiederum wird mit dem effektiven Start (*ActualStart*) und der Grad der Fertigstellung (*Completion*) errechnet (buildingSMART International, 2022).

Da die technischen Voraussetzungen für die Arbeit mit Low-Code Plattformen gering sind, zeigt sich insgesamt, dass Low-Code-Plattformen vielversprechende Chancen haben, um als Instrument flächendeckende Anwendung zu finden. Indem Low-Code-Plattformen mit vorgefertigten Bausteinen die Möglichkeit bieten, kleine bis grosse Arbeitshilfen zu programmieren, wird eine grosse Akteursgruppe angesprochen, welche Lücken von IT-Fachpersonal in nützlicher Zeit zu schliessen versucht. Einige Experten weisen auf die erfolgreiche Zukunft dieser Plattformen hin und heben ihre Hauptvorteile hervor. (Bock & Frank, 2021, p. 733; Sanchis et al., 2019, p. 3; Shah, 2020; Vincent et al., 2020, p. 1). So wird innerhalb kurzer Zeit eine Low-Code-Lösung entwickelt, die die meisten manuellen Verarbeitungsschritte eliminiert und so die Verarbeitungszeit verkürzt, während gleichzeitig ein zufriedenstellender Überblick erhalten bleibt (Vikebø & Sydvoid, 2019, p. 2). Zugleich wird durch die direkte Einbindung der Endanwender eine Stärkung der Benutzer erzielt, da sie ihre Bedürfnisse unmittelbar in den Entwicklungsprozess einbringen und so die Lösungen passgenau ausgestalten können (Bock & Frank, 2021, p. 733). Eine Forrester-Studie belegt eine Zunahme der Schnelligkeit um das fünf- bis Zehnfache des Entwicklungsprozesses auf Low-Code-Plattformen, was gleichzeitig zu erheblichen Kosteneinsparungen führt (Richardson & Rymer, 2016, 2014). Durch die Reduzierung der Komplexität können auch Fachexperten ohne technologische Vorkenntnisse zu App-Entwicklern werden und ihre tiefgehenden Geschäftskennntnisse direkt in den Entwurf einbringen (Waszkowski, 2019, pp. 376–381). Trotzdem richten sich Low-Code-Anwendungsplattformen in erster Linie an Vollzeit- und Teilzeientwickler. (IBM, 2021)

Doch die Datendurchgängigkeit und flexible Schnittstellen sind zentrale Herausforderungen moderner Low-Code-Umgebungen. Zugleich bietet Low-Code die Chance, bestehende Funktionalitätslücken in Altsystemen zu schliessen und die Widerstandsfähigkeit gegenüber Updates oder unerwarteten Systemunterbrüchen zu erhöhen. Denn fehlende Schnittstellen können meist schnell ergänzt und zentral gewartet werden. Die Sicherstellung durchgängiger Datenflüsse über mehrere Systeme hinweg ist nach wie vor ein zentrales Thema mit enormem Potenzial für die Zukunft. Grundlegend wird ein sorgfältiges Rollen- und Rechte management gefordert, diese Vielfalt zu beherrschen ist. Im digitalen Lebenszyklus von Baudaten zeigt sich eine erhebliche Diskrepanz zwischen den langfristigen Anforderungen an die Daten-Qualität, das Informationen über Jahrzehnte hinweg vorhalten muss, und der kurzzyklischen Perspektive der IT. (Linderoth et al., 2018) Während branchenspezifische Low-Code-Anwendungen aufgrund ihrer besonderen Anforderungen häufig an die Grenzen der Standardisierung stossen.

Standardisierungen auf der Baustelle sind laut den befragten Experten am schwierigsten, weil jede Baustelle andere Anforderungen mit sich bringt. Die Realität zeigt, dass Low-Code vor allem für kleine Automatisierungen, welche für den interne Bereich genutzt werden, zur Anwendung kommen. Trotzdem gibt es aber auch Exemplarische Anwendung für die Verwendung auf der Baustelle und die Planung. Diese machen Sinn, weil dort ein grosses Potential an Einsparung schlummert, was der Hauptgrund ist, um neue Technologien zu nutzen. Dazu gehört auch das Messen des Baufortschrittes, da dieser unmittelbar mit der Vergütung der Bauarbeiten verbunden ist. Auf dem Markt gibt es bereits einige Apps, welche den Baufortschritt und weiteres wie alle Details bezüglich Bauaktivität, Materialien, Maschinen, Anlagen, Arbeitsbedingungen und Problemen aufnehmen können. Jedoch fehlt der direkte Bezug zum digitalen Bauwerksmodell, spezifisch zum IFC-Datenschema, wodurch der OpenBIM-Ansatz nicht berücksichtigt wird (smapOne, 2025).

Auch andere beschreiben, dass Low-Code-Plattformen wie die Microsoft Power Platform IFC-Dateien nicht direkt lesen und somit nicht ohne Weiteres integrieren lassen (buildagil, 2025).

Zusammenfassend lässt sich aus der Recherche ableiten, dass die Low-Code Technologie im Zusammenhang mit IFC und dem digitale Bauwerksmodelle mit einer Problemumgehung (workaround) umsetzen lassen würde. Dabei könnten die relevanten Daten aus der IFC-Datei mithilfe eines Python-Codes extrahiert und in einer CSV-Datei abgelegt werden. Diese könnte wiederum als Grundlage für die Erstellung eines Excel-Dokument dienen, welche als Daten-bank agiert und anschliessend in der Applikation durch den Bauleiter vor Ort bearbeitet werden könnte. Die bearbeiteten Informationen könnten danach wiederum mit einem Python-Code in die IFC-Datei zurückgeschrieben und sind somit für alle an der Projekt Beteiligten zugänglich gemacht werden.

3 Beschreibung des Forschungsbedarfs

In Bezug auf Low-Code-Entwicklungen wird leider eine sehr unkritische Haltung eingenommen (Bock & Frank, 2021, p. 734). Die Forschung über die Low-Code-Technologie im Zusammenhang mit IFC und dem digitalen Bauwerksmodell ist noch nicht dokumentiert und bisher unzureichend. Bisherige Studien fokussieren sich ausschliesslich auf andere Branchen und deren Verwendung von Low-Code. So wurde bislang nicht nach der Schliessung von Schnittstellen-Lücken geforscht. Dabei zeigt die Erfahrung, dass digitale Lösungen oder Anwendungen, die mit Low-Code-Plattformen entwickelt wurden, keine neuen Dateitypen generieren. Stattdessen nutzen sie bestehende Formate, um die Interoperabilität zwischen verschiedenen Softwareprogrammen zu erleichtern. Dadurch wird die fragmentierte Softwarelandschaft besser integriert und die Bildung neuer Datensilos vermieden. (Martinez & Cisterna, 2023, p. 204)

Weiter benötigen die meisten Unternehmen massgeschneiderte Software für einen bestimmten Anwendungsfall, um ihre Aufgaben möglichst effizient und effektiv erledigen zu können (Weidmann et al., 2024, p. 7). Die Daten, die digitalisiert und mit Low-Code verarbeitet werden, könnten weiter genutzt und in ein umfassenderes System integriert werden. So könnten beispielsweise vor Ort erfasste Daten mit dem BIM Industry Foundation Classes (IFC)-Schema verknüpft werden, um die im Modell vorhandenen Daten zu erweitern. Da Low-Code in der Bauindustrie noch relativ neu ist, bieten solche Schnittstellen einen interessanten Bereich für weitere Forschungsarbeiten.

Zudem sind einzelne Elemente wie etwa ein Terminprogramm in der Planung oder die Attribuierung vom Baufortschritt im IFC bereits vorhanden. Stand heute hat aber niemand diese Themen im Schweizer Baukontext und den geltenden Gesetzen und Normen zusammengeführt. (Martinez & Cisterna, 2023, p. 205)

Die wissenschaftliche Relevanz besteht daher in der Erforschung einer ganzheitlichen Integration vom IFC-Datenschema in der Baubranche mithilfe von Low-Code. Ziel ist es, wiederkehrende Arbeitsabläufe zu vermeiden und fehlende Schnittstellenlücken zu schliessen. Der Baufortschritt soll dabei über einen Wert abgefragt und bearbeitet werden können während das jeweilige Bauteil gleichzeitig visualisiert werden kann (Martinez, 2024b).

Mit der Beantwortung nachfolgender Forschungsfrage soll diese Wissenslücke geschlossen werden:

- Wie könnte die Low-Code Technologie im Zusammenhang mit IFC und dem digitale Bauwerksmodelle in den Arbeitsalltag integriert werden?

Zusätzlich mit den Teilfragestellungen

zu Chancen und Herausforderungen:

- Welche Chancen bietet der Einsatz von Low-Code Technologie im Vergleich zu herkömmlichen Programmiermethoden im Bauwesen?
- Was sind die Herausforderungen und möglichen Risiken bei der Integration von Low-Code-Technologien in den Arbeitsalltag im Bauwesen?

für konkrete Anwendungsszenarien:

- Welche konkreten Anwendungsbeispiele gibt es bereits für den Einsatz von Low-Code-Technologien im Bauwesen?
- Wie können typische Arbeitsabläufe im Bauwesen durch die Integration von Low-Code Technologie und IFC-Daten verbessert werden?

für die technische Umsetzung:

- Welche technischen Voraussetzungen müssen erfüllt sein, um Low-Code-Plattformen mit IFC-kompatiblen Softwarelösungen zu integrieren?
- Welche Low-Code-Plattformen unterstützen die Arbeit mit IFC-Daten und digitalen Bauwerksmodellen?

4 Methodik

Als Grundlage dieser Thesis dienen umfassende theoretische Grundlagen, die mittels systematischer Literaturrecherche erarbeitet wird. Darauf aufbauend soll das Wissen in diesem Forschungsgebiet gezielt erweitert werden (Webster & Watson, 2002). Ergänzend dazu werden Expertengespräche durchgeführt, um gezielt relevante Faktoren für die praktische Umsetzung zu ermitteln. Zur besseren Strukturierung erfolgt eine Eingrenzung der beteiligten Akteursgruppen in: Unternehmen, Entwickelnde und Nutzende. Deren jeweilige Interessen und Ansprüche werden als zentrale Bezugsgrößen für die weitere Analyse herangezogen. Geografisch bezieht sich die Thesis primär auf die Schweiz und richtet den Fokus auf die ausführende Baubranche. Sie stützt sich dabei auf den aktuellen Forschungsstand sowie auf anerkannte Regeln der Baukunde. Zusätzlich werden gezielte und themenspezifische Internetrecherchen durchgeführt, um bestehende Wissenslücken zu schliessen. Das zusammengefasste Literaturrecherche-Protokoll ist ausführliche Rechercheprotokoll ist im Anhang A0 nachzulesen.

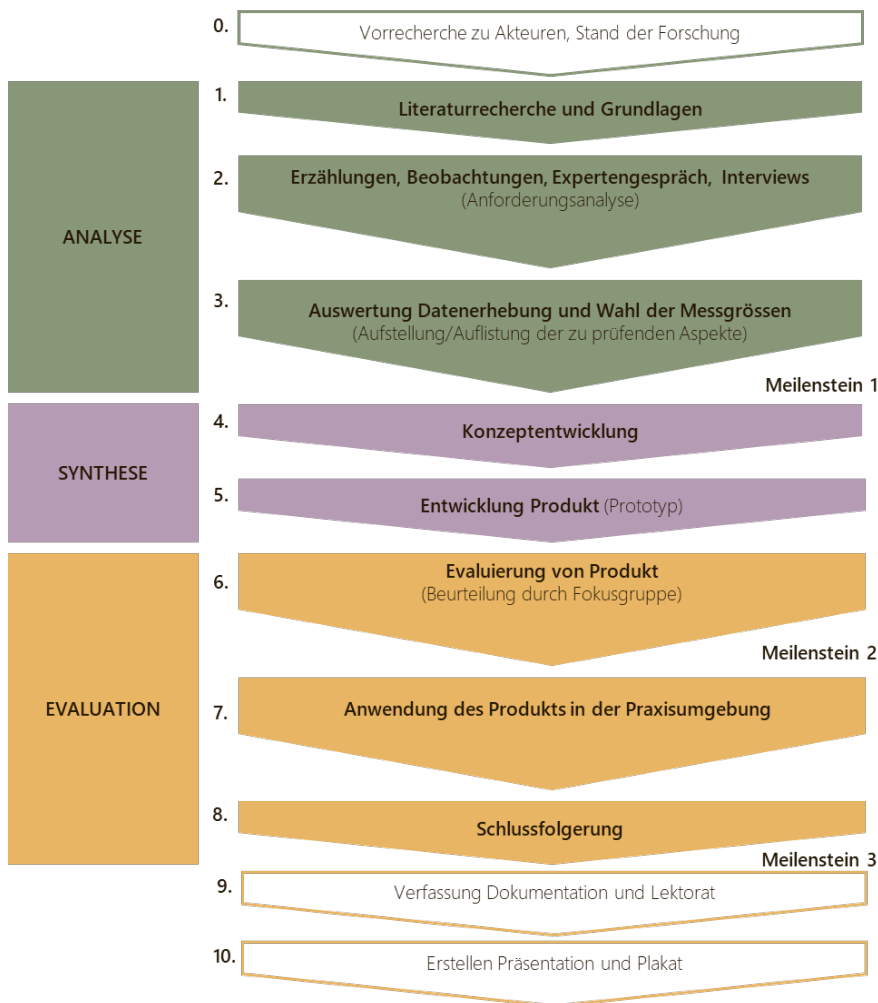
Der geplante Ablaufplan für das Vorgehen wird in den Phasen drei und sechs entgegen dem Proposal marginal angepasst. Dies ist dem Umstand geschuldet, dass keine Studien zu aktuellen Anwendungsszenarien in der Baubranche mit Low-Code-Applikationen zu finden sind. Weshalb aktuelle Anforderungen mit der Methode der Aktionsforschung erfragt werden, um die Bedürfnisse für eine Applikation zu erstellen.

In der Phase drei wurden die Messgrößen nicht mit dem Praxispartner definiert, sondern aufgrund der Ergebnisse aus den qualitativen Experteninterviews aus dem Bauhaupt- und Baunebengewerbe.

Der Prototyp wurde eigenständig aufgebaut und mittels Fokusgruppe mit dem Praxispartner validiert. Es wurde jedoch auf eine Überarbeitung und eine 2. Validierung in der Phase sechs verzichtet, da dieser 1. Prüfschritt bereits genügend Erkenntnisse lieferte und die Anpassungen ausschliesslich der Verbesserung vom Userinterface dienen würden ohne einen Einfluss auf das Zusammenspiel vom IFC mit der MS-Powerapp mittels Low-Code zu haben.

Aus den Schlussfolgerungen zum Stand der Forschung wird das Forschungsdesign dieser Thesis wie folgt abgeleitet:

4.1 Forschungsdesign / Forschungsmethodik



Die Master-Thesis wird gemäss dem in Abbildung 22 dargestellten Erstellungsprozess und nach den folgenden Schritt- und Methodenerläuterungen durchgeführt. Zur Transparenz und Steuerung der Forschungsdurchführung sind drei Meilensteine vorgesehen.

Abbildung 22, Ablauf zur Vorgehensweise (eigene Abbildung)

4.2 Analyse

Phase 1 und 2 | theoretische und praktische Datenerhebung

Die Analyse stützt sich einerseits auf die Literaturrecherche, andererseits auf qualitative Interviews, die entlang zentraler Forschungsfragen geführt wurden und anschliessend strukturiert wurden. Mithilfe dieser strukturierten, kategorischen Analyse soll ein differenziertes Bild der Einsatzbedingungen und Wirkungen von Low-Code-Anwendungen aus Anwendersicht entstehen.

Um zu untersuchen, welche Formen der organisatorischen und technischen Integration geeignet sind, um das innovative Potenzial der Low-Code-Technologie zu nutzen, wurde das Wissen unter anderem durch neun Interviews in verschiedenen Unternehmen erhoben. Diese Unternehmen wurden ausgewählt, weil erstens der Einsatz von Low-Code in der Baubranche schrittweise erfolgt ist, ohne dass dazu bisher Publikationen erschienen sind. Zweitens war es für diese Arbeit wichtig, ein breites Bild darüber zu gewinnen, welche Aspekte im Alltag von den Ausführenden als besonders relevant angesehen werden.

Dafür wurden verschiedene ausführende Unternehmen der Bauindustrie interviewt, um Ineffizienzen bei der Datenerfassung und -verarbeitung im Rahmen der Implementierung einer digitalen Low-Code-Anwendung zu identifizieren. Die verschiedenen Akteure wurden weiter mit den damit verbundenen Chancen und Herausforderungen, konkreten Anwendungsszenarien und den Voraussetzungen für die technische Umsetzung befragt. Die Expertengespräche wurden dabei auf der Grundlage eines Interviewleitfadens geführt (siehe Anhang A11). Hierbei handelt es sich um eine qualitative Befragungsmethode, die sich durch offene Fragen auszeichnet. Diese Fragen

wurden so formuliert, dass die Befragten dazu angeregt werden, ihre Gedanken zu den jeweiligen Themen frei mitzuteilen (DiCicco-Bloom & Crabtree, 2006).

Die Interviews wurden aufgezeichnet und stichwortartig dokumentiert. Die Interviews wurden auf Schweizerdeutsch geführt und anschliessend ins Hochdeutsch übersetzt. Die Aufnahmen wurden nach Abschluss der Transkription gelöscht. Die zu Interviewenden erhielt ein Informationsschreiben, in dem die Ziele des Interviews dargestellt wurden. Zusätzlich unterzeichneten sie eine Einverständniserklärung, in der sie der Aufzeichnung, Transkription und Verwendung der Interviewsitzung für unsere Forschung zustimmten. Um die Daten für die Analyse vorzubereiten wurden die Antworten zur späteren Verwendung nach Themen sortiert.

Phase 3 | Auswertung und Definition der Messgrössen

In der Phase der Analyse wurde anhand der gesammelten Informationen aus den Interviews mit den Bauexperten (Fachexpert:innen) einerseits die Auflistung der geforderten Informationen für die Entwicklung einer Applikation erstellt.

Andererseits wurde aus diesen Interviews Messgrössen abgeleitet und eine Auswertungsliste erstellt, welche eine Übersicht über Messgrössen im Zusammenhang mit der Einführung und Nutzung von Low-Code-Technologien in verschiedenen Unternehmens- und Projektkontexten liefert. Die Detaillierte Liste kann im Anhang A8 eingesehen werden. Im Fokus stehen dabei hauptsächlich qualitative Merkmale.

Bei der Festlegung dieser Messgrössen wurde ein qualitatives Auswertungskonzept angewendet, das sich gut mit einer qualitativen Inhaltsanalyse nach Mayring umsetzen lässt (Mayring, 2022). Dabei wird sowohl systematisch und theoriegeleitet (deduktiv) als auch offen für neue Kategorien (induktiv) vorgegangen. Dies nach dem Schema:

1. Material auswählen
2. Richtung der Analyse festlegen
3. Form der Inhaltsangabe auswählen
4. Ergebnisse interpretieren
5. Gütekriterien sicherstellen

Diese Technik unterstützt die Lösung praktischer Probleme und erweitert gleichzeitig gemeinsam das wissenschaftliche Wissen durch die Zusammenarbeit verschiedener Akteure. Dabei schafft die Anwendung der Aktionsforschung einen geeigneten Rahmen für die Durchführung angewandter Forschung in der Bauindustrie und fördert die Zusammenarbeit zwischen Wissenschaft und Praxis (Azhar et al., 2009).

Zusätzlich werden qualitative Einschätzungen zu Chancen und Herausforderungen der Technologie dokumentiert. Hierzu zählen Aspekte wie: die vereinfachte Programmierung, die schnelle Umsetzung von Prototypen, die dynamische Anpassungsfähigkeit an Nutzerbedürfnisse sowie potenzielle Einschränkungen durch fehlende Funktionen oder spezifische Umsetzungsprobleme. Die Aussagen spiegeln eine differenzierte Bewertung der Technologie durch die Praxis wider und liefern wertvolle Anhaltspunkte für weiterführende Analysen zur Effektivität von Low-Code-Lösungen im betrieblichen Alltag. Die Ergebnisse sind in Kapitel 5 nachzulesen.

Meilenstein 1, Definition Messgrössen: Besprechung und Fixierung der Messgrössen sowie Evaluationskriterien mit den Thesis-Experten und der Thesis-Begleiterin.

4.3 Synthese

In der Entwicklungsphase wurde die SCRUM-Methode angewendet. Das bedeutet, dass die Entwicklung in mehrere Sprints unterteilt wurde, die jeweils auf eine Veröffentlichung hinführen. Während dieser Sprints wird regelmässig überprüft, ob die gesetzten Ziele erreicht wurden und ob Anpassungen notwendig sind. Kurz gesagt handelt es sich um einen agilen Entwicklungsansatz, bei dem der Kunde und das Endprodukt eine zentrale Rolle spielt. Ein Sprint kann per Definition nicht verschoben werden, da der Zeitplan fix vorgegeben ist. (Christensen, B. H., 2018; Vikebø & Sydvold, 2019, p. 24).

Während dem Erarbeitungsprozess kann es vorkommen, dass durch die Interaktion im Entwicklungsprozess neue Ideen entstehen, welche die ursprünglich geplanten weiteren Bearbeitungsschritte als unnötig oder überflüssig erscheinen lassen. Ebenso kann es sein, dass sich neue, sinnvollere Richtungen abzeichnen. Daher ist es eine wichtige Fähigkeit einer agilen Organisationsstruktur, solche Vorschläge im Verlauf des Prozesses zu berücksichtigen. Diese Iterationen werden solange wiederholt bis das Ziel erreicht und von den Kunden freigegeben ist (Vikebø & Sydvold, 2019, p. 24).

Phase 4 | Konzeptentwicklung

Basierend auf den erhobenen Metriken wurde im nächsten Schritt ein umfassendes Konzept für die geplante Anwendung entwickelt. Die in Phase 3 erarbeiteten Ergebnisse zu den Anforderungen, Wünschen und Bedürfnissen der zukünftigen Nutzer wurden dafür weiter ausgearbeitet, um eine bestmögliche User Experience zu gewährleisten.

Das Konzept umfasst sowohl funktionale als auch gestalterische Aspekte. Die funktionalen Aspekte beziehen sich auf die technischen Anforderungen und die Leistungsfähigkeit der Anwendung. Die gestalterischen Aspekte umfassen das Design, die Benutzerführung und die visuelle Gestaltung. Durch diesen integrativen Ansatz wird eine Lösung angestrebt, die nicht nur benutzerfreundlich, sondern auch effizient und intuitiv bedienbar ist.

Phase 5 | Prototypen

Auf Grundlage des Konzeptes wird mit dem Programm Power-Apps von Microsoft eine Prototyp-Applikation entwickelt. Diese ermöglicht den Abruf von Informationen zu einem Bauteil aus einer IFC-Datei, kann diese visuell und verknüpft darstellen, die Informationen bearbeiten und sie wieder in die IFC-Datei zurückführen. Dabei werden die in Meilenstein 1 definierten Bedürfnisse berücksichtigt.

4.4 Evaluation

Die Verfasserin verwendet Aktionsforschung zur Erarbeitung des Endproduktes. Diese Methode folgt dem Ansatz einer sich wiederholenden Spirale von drei Schritten: 1. Planung, 2. soziale Intervention im Feld und 3. Reflexion der Ergebnisse der Intervention. (Kurt Lewin, 1953)

Phase 6 und 7 | Beurteilen und Anwendung des Produktes in der Praxisumgebung

Die Evaluierung erfolgt mittels Expertengesprächen (Fokusgruppe) mit der Praxispartnerin, welche sowohl als Unternehmerin, aber auch als Entwickelnde und Nutzenden der Applikation diese bewerten. Die Beurteilung orientiert sich an vordefinierten Kriterien.

Die Fokusgruppe dauerte eine Stunde und bestand aus drei Teilen:

1. Warum, Ziel Vision:

Als erstes wurden das Konzept und der bisherige Arbeitsstand der Applikation mit den entsprechenden Datenflüssen gemäss Abbildung 48 erklärt. Dieser Teil nahm ca. 15 Minuten in Anspruch.

2. Diskussion:

Anschliessend wurden während rund 40 Minuten Fragen zu den jeweiligen Forschungs- bzw. Teilfragestellungen diskutiert. Als gemeinsame Wissensgrundlage diente dabei der Prototyp. Die Diskussion umfasste folgende Themen:

- **Chancen:** Was sind die Chancen im Vergleich zu herkömmlichen Programmiermethoden?
- **Herausforderungen und mögliche Risiken: Wo könnte Low Code an seine Grenzen stossen und welche Risiken gilt es zu beachten?**
- **Typische Arbeitsabläufe im Bauwesen:** Wie laufen Prozesse derzeit in eurem Alltag ab? An welchen Stellen könnte Low Code sinnvoll eingebunden werden, um Abläufe zu optimieren?
- **Technische Voraussetzungen:** Welche Infrastruktur oder Systemanbindungen sind notwendig, damit eine Low-Code-Plattform reibungslos funktioniert? Gibt es spezielle Anforderungen an Sicherheit oder Schnittstellen?
- **Auswahl geeigneter Low-Code-Plattformen:** Mit welchen Tools habt ihr bereits Erfahrungen und welche würdet ihr für unser Szenario empfehlen?

3. Umfrage zu Beurteilungskriterien und Gewichtungen (Nutzwertanalyse):

Der Abschluss dauerte ca. 10 Minuten. In diesem Rahmen bearbeiteten die Teilnehmenden einzeln einen digitalen Mentimeter-Fragebogen, ein Instrument zur Durchführung interaktiver Umfragen. Die detaillierten Fragen sind dem Anhang A9 zu entnehmen. Die Auswertung ist in Kapitel 5.1.4 sowie ergänzend, umfassend im Anhang A10 dargelegt.

Meilenstein 2, Produktevaluation: Vergleich des Produkts mit den Anforderungskriterien.

Phase 8 | Schlussfolgerung

Auf der Basis der gesammelten Erkenntnisse über das bewertete Produkt wird eine Schlussfolgerung zum Produkt, aber auch zu Chancen und Herausforderungen bei der Verwendung von Low-Code-Applikationen formuliert.

Meilenstein 3, Schlussfolgerung: Diskussion der Chancen und Herausforderungen während dem Erarbeitungsprozess sowie des Ausblicks für zukünftige Forschungen zusammen mit der Thesis Begleiterin.

5 Ergebnisse

Aus der Literaturrecherche, dem Dokumentenstudium und den Expertengesprächen wurden für die zu klärende Hauptfragestellung folgende Ergebnisse abgeleitet: Es wurde erläutert, wie sich Low-Code-Technologien im Zusammenspiel mit IFC-basierten digitalen Bauwerksmodellen nahtlos in den Arbeitsalltag der Baupraxis integrieren lassen. Um dieser Forschungsfrage systematisch nachzugehen, gliedern sich die Resultate in fünf Themenbereiche:

Im **ersten Teil** werden die notwendigen infrastrukturellen und softwareseitigen Voraussetzungen herausgearbeitet, die eine reibungslose Integration eines digitalen Bauwerksmodells auf einer Low-Code-Plattform ermöglichen. Dazu wird eine mehrdimensionale Betrachtung des Baufortschritts als exemplarische Basis verwendet (siehe Kapitel 5.1). Das Kapitel beinhaltet zusätzlich die praktische Umsetzung des Prototyps mit der Low-Code-Technologie (siehe Kapitel 5.1.3). Abgeschlossen wird dieses Kapitel mit der Durchführung einer Fokusgruppe, in der die noch offenen Fragen anhand der eigens aufgebauten Low-Code-Applikation diskutiert werden (siehe Kapitel 5.1.4).

Konkrete Anwendungsszenarien werden im **zweiten Teil** erörtert. Anhand bereits existierender Beispiele wird analysiert, in welchen Bereichen und Prozessen Low-Code-Lösungen heute eingesetzt werden. Weiter wird aufgezeigt, wie sich typische Arbeitsabläufe von der Fortschrittsdokumentation bis zur Materialverwaltung durch die Verbindung von Low-Code-Tools mit IFC-Daten optimieren lassen (siehe Kapitel 5.2).

Der **dritte Teil** beleuchtet die Chancen und Herausforderungen. Hier gilt es zu klären, welche Vorteile Low-Code-Ansätze gegenüber klassischen Programmiermethoden im Bauwesen bieten und welchen Risiken bei ihrer praktischen Anwendung zu begegnen ist (siehe Kapitel 5.3).

Im **vierten Teil** wird ein Produkt erörtert, welches bereits die digitale Erweiterung bestehender Arbeitsabläufe mittels Low-Code ermöglicht (siehe Kapitel 5.4).

5.1 Integration eines digitalen Bauwerksmodells auf einer Low-Code-Plattform

5.1.1 Exemplarisches Use-Case | Baufortschritt im Schweizer Kontext

Die Grundlagen für die Erfassung und Steuerung des Baufortschritts liegen im Schweizer Bauwesen in unterschiedlichen Einzeldisziplinen vom Terminprogramm bis hin in der SIA 102:2020-geregelten Baujournal. Bislang fehlt jedoch eine ganzheitliche Zusammenführung dieser Aspekte im Kontext mit Low-Code. Eine mehrdimensionale Beurteilung des Baufortschritts muss daher die planerische Vorgabe, die digitale Modellierung, die normativen Rahmenbedingungen, die praktische Dokumentation sowie den Soll-Ist-Vergleich berücksichtigen.

Durch die Verknüpfung von Terminplanung, 4D-BIM-Daten, normierter Dokumentation und fortlaufender Abweichungsanalyse lässt sich im Schweizer Bauwesen eine transparente, mehrdimensionale und rechtskonforme Beurteilung des Baufortschritts ermöglichen. Eine App zur ganzheitlichen Baufortschrittserfassung verbindet diese in der Abbildung 23 ersichtlichen Komponenten in einer Low-Code-Umgebung. Diese enthält die nachfolgenden Elemente:

- (1) **Balken-Diagramm** für die Planung der Baustelle (Soll).
- (2) **Delta** = Differenz zwischen Balkendiagramm (Soll) zu Baujournal (Ist).
- (3) **Datenbank** als zentraler Speicherort für alle Beteiligten.
- (4) **Feedback der Zeiten** als aktuelle Ist-Aufnahme auf der Baustelle.
- (5) **Baujournal** als Rahmen zur alltäglichen Dokumentation von anfallenden Informationen auf der Baustelle.
- (6) **SIA-Normen** definieren die Pflichten des Bauleitenden.
- (7) Das **digitale Bauwerksmodell (DBM)** vereint Informationen aus der Planung wie aus dem Baumanagement.
- (8) Die standardisierte Struktur des **IFC-Datenschemas** gibt den spezifischen Ablageort der Informationen zur Bauplanung/-kontrolle im digitalen Bauwerksmodell vor.

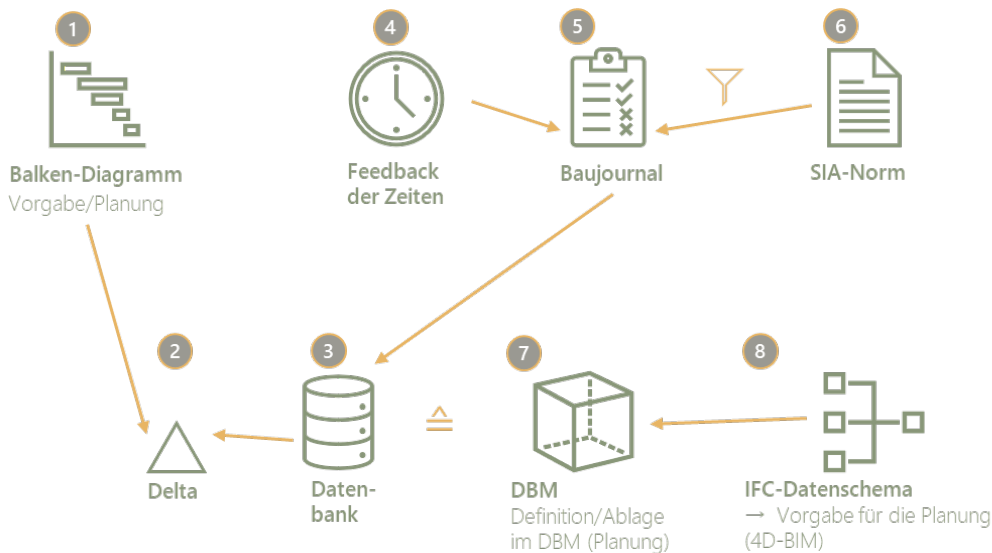


Abbildung 23; Komponenten zur mehrdimensionalen Betrachtung des Baufortschritts (eigene Abbildung)

5.1.2 Vorgehensprozess zur Integration eines IFC in einer Low-Code-Plattform

Derzeit gibt es jedoch vor allem Insellösungen, welche die Bauplanung unabhängig von einem standardisiertem Datenformat und innerhalb ihres Systems ermöglichen. (beispielsweise Messerli Protokoll) Die Daten sind somit nur für andere Lizenznehmer, meist aber nicht für alle Planenden zugänglich. Eine App zur ganzheitlichen Baufortschrittserfassung soll diese Komponenten in einer Low-Code-Umgebung miteinander verbinden und so eine durchgängige Systematik ermöglichen. Der Aufbau vom Prototyp gliedert sich grob in die folgenden drei Teile:

1. **Anforderungsdefinition:** Strategische Ziele, Benutzerbedürfnisse und Unternehmensstrategie
2. **Fachlich-konzeptuelle Modellierung:** Strukturierung von Funktionalität, Klassifikation nach BIM-Handbuch, Typische BIM Modellkoordination, BIM-Referenzarchitektur
3. **Technische Implementierung:** Strukturierung der Softwarebausteine

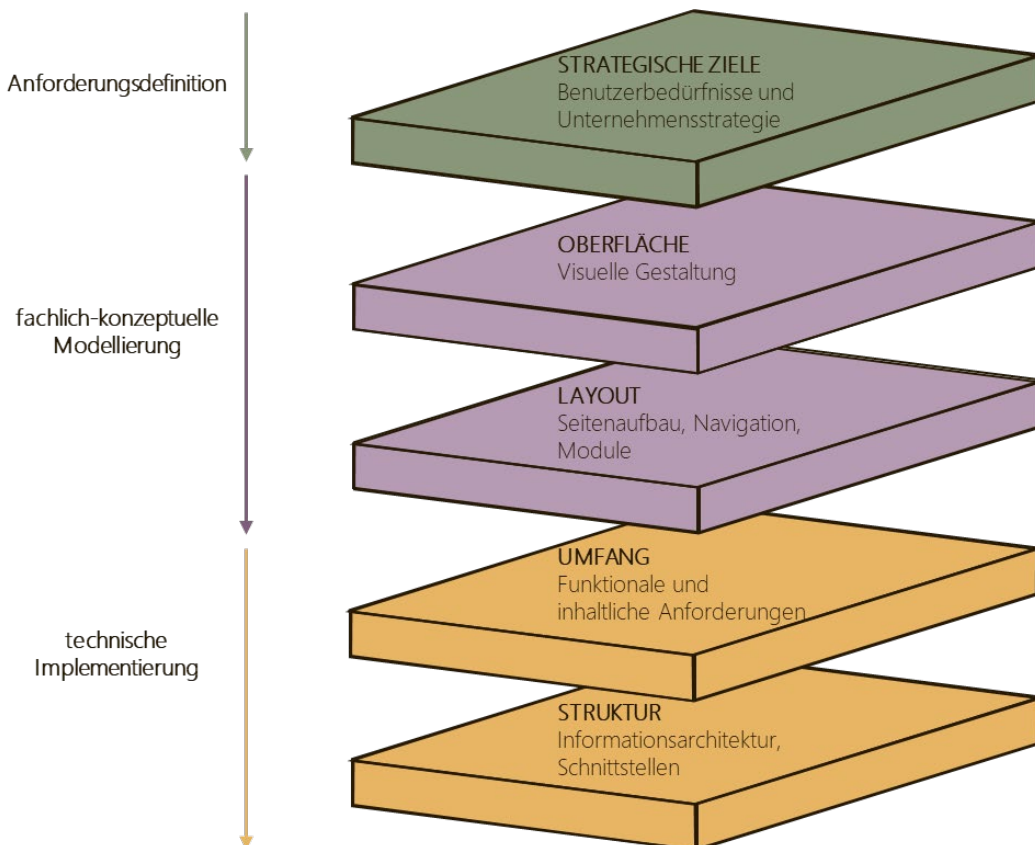


Abbildung 24; Vorgehensprozess zur Integration eines IFC in einer Microsoft Power App (eigene Abbildung)

5.1.3 Umsetzung Prototyp «Integration eines IFC in einer Microsoft Power App»

Der exemplarische Prototyp ist nach dem im Kapitel 5.1.2 und der Abbildung 24 beschriebenen Vorgehen entwickelt.

5.1.3.1 Anforderungsdefinition

► STRATEGISCHE ZIELE

Das Ziel der Entwicklung eines Business Case besteht nicht darin, den genauen Wert der Investition zu berechnen, sondern vielmehr zu zeigen, dass die Investition sinnvoll ist. In dieser Phase erfährt die Organisation, welche Wertoptionen es gibt und wie sich die Technologie auf ihr Betriebsmodell auswirken wird. Ein funktionales Design, die Berücksichtigung technischer Aspekte als kohärente Logik und die Architektur sind für eine erfolgreiche Umsetzung unerlässlich (Vikebø & Sydvold, 2019, p. 15).

Unternehmensstrategie

Aus den Interviews geht hervor, dass sich die Branche eine mit Low-Code entwickelte App wünscht, die auf der Baustelle zur Erfassung des Baufortschritts eingesetzt werden kann (Blessing & Asairatnam, 2025; Bosshart, 2025; Pfister, 2024). Der Baufortschritt soll dabei über einen Wert erfasst und bearbeitet werden können, wobei das jeweilige Bauteil visualisiert werden soll (Martinez, 2024b).

Ziele der Applikation

- **Baufortschritt:** Abrufen und vergleichen Stand zur Planung sowohl im Büro als auch auf der Baustelle.
- **Transparenz:** Automatische Einhaltung aller SIA-Journal-Vorgaben und Exportfunktionen zur Abrechnung (SIA 118, SIA 102).
- **Interoperabilität:** Dank Speicherung in Standardisiertem Format, unabhängig von Software abrufbar.
- **Mobiler Mehrwert:** Die Indoor-/Outdoor-App für Tablet oder iPad ermöglicht die lückenlose Erstellung eines Baujournals direkt auf der Baustelle.

Benutzerbedürfnisse

Auf der Baustelle zählt jede Minute, daher muss sich eine App nahtlos in den rauen Alltag einfügen. Die Nutzenden erwarten, mit wenigen Klicks Informationen abrufen oder weitergeben zu können, beispielsweise das Baujournal oder den aktuellen Bearbeitungsstand. Klare und grosse Symbole sowie eine einfache Navigation sind wichtig, damit niemand lange suchen muss. Am Ende geht es darum, den Arbeitsfluss zu erleichtern, Fehlerquellen zu vermeiden und für alle Beteiligten mehr Transparenz zu schaffen (Blessing & Asairatnam, 2025; Bosshart, 2025; Pfister, 2024).

Vorgehensprozess zur Baufortschrittskontrolle (heute)

Der bestehende Prozess hat sich über lange Zeit eingependelt und gilt als bewährt. Dennoch führt er bei agilen Projektsteuerungen bei allen Beteiligten zu Unbehagen, da Rückstände meist zu spät erkannt werden. In einem traditionellen Bauprozess beginnt das Projekt mit der Erstellung einer Planung, auf deren Grundlage ein Terminplan ausgearbeitet wird. Dieser dient als Richtschnur für die nachfolgenden Bautätigkeiten. Während der Bauausführung wird in regelmässigen Abständen ein manueller Abgleich zwischen dem geplanten und dem tatsächlichen Baufortschritt durchgeführt. Diese Kontrolle basiert dabei oft auf Beobachtungen, Notizen oder Berichten und ist stark von individueller Dokumentation und Erfahrung abhängig. Der Prozess folgt dabei nachfolgenden Auflistungen beziehungsweise anhand dem Flussdiagramm in der Abbildung 25:

Start

1. **Geplante Bautätigkeit erfassen**
Die Bautätigkeit wird geplant und in einem Terminprogramm erfasst.
2. **Tatsächliche Bautätigkeit erfassen**
Die aktuellen Daten werden auf der Baustelle erfasst.
3. **Manueller Vergleich Soll – Ist**
Die geplanten und tatsächlichen Bautätigkeiten werden verglichen.
4. **Abweichung vorhanden?**
Gibt es eine Abweichung zwischen dem Bauprogramm und der Realität? Entscheidung mit Ja oder Nein
5. **Abweichung analysieren**
Die Abweichung wird analysiert und der Grund dafür gesucht.

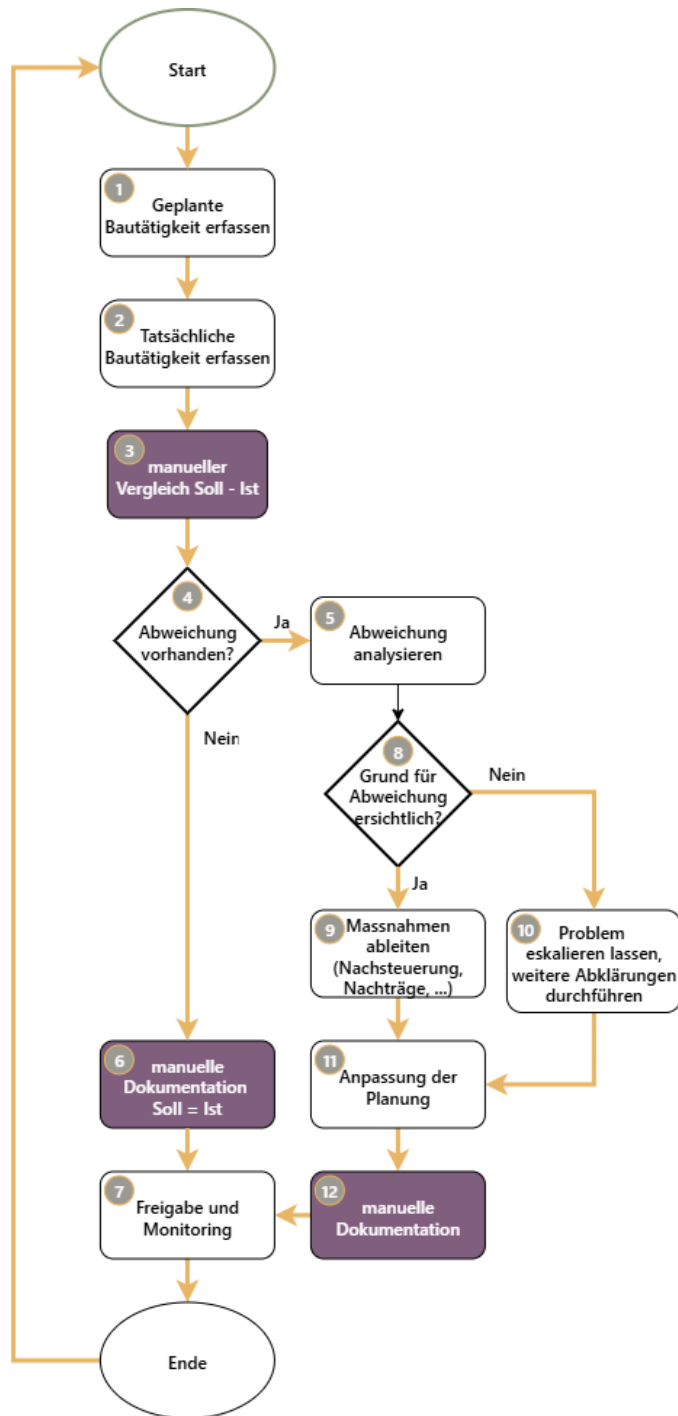


Abbildung 25; Vorgehensprozess zur Baufortschrittskontrolle (heute)

6. **Manuelle Dokumentation:** Soll = Ist
Falls keine Abweichung besteht, wird dokumentiert, dass Soll = Ist entspricht.
 7. **Freigabe und Monitoring**
Der Prozess geht in die Überwachung über.
 8. **Grund für Abweichung ersichtlich?**
Ist der Grund für die Abweichung ersichtlich? Entscheidung mit Ja oder Nein. Beispielsweise weniger Arbeiter auf der Baustelle als geplant.
 9. **Massnahmen ableiten**
Konkrete Gegenmassnahmen werden beschlossen (z.B. Nachsteuerung, Nachträge ...).
 10. **Problem eskalieren lassen, weitere Abklärungen durchführen.**
Falls der Grund unklar ist, wird das Problem eskaliert und es folgen weitere Abklärungen. Danach Rückführung zu 11 (Anpassung der Planung).
 11. **Anpassung der Planung**
Aktualisierung und Korrektur der ursprünglichen Planung. Dies kann das Terminprogramm betreffen aber auch die geometrische Planung.
 12. **Manuelle Dokumentation**
Die Anpassungen und Massnahmen werden dokumentiert. Danach Rückführung zu 7 (Freigabe und Monitoring)
- Ende**
Mit dem Ende des Ablaufs beginnt er wieder von vorne bis das Ende der Bauarbeiten erreicht ist.
Bisher erstreckte sich der Iterationszyklus für den Soll-Ist-Abgleich jeweils vom Abschluss einer Phase bis zum nächsten Phasenabschluss oder wurde bei Projektänderungen beziehungsweise bei Unstimmigkeiten durch den Bauherrn ausgelöst.

Dies hat zur Folge, dass Korrekturen nicht nur grössere Anpassungen erfordern, sondern auch aufwändige und teilweise langwierige Genehmigungsprozesse nach sich ziehen. Die späte Einwirkung schränkt den Handlungsspielraum dabei erheblich ein, wodurch leider vielfach mangelhafte Kompromisslösungen angewandt werden.

Zusätzlich können solche verspäteten Eingriffe die Unsicherheiten im Projektteam verstärken. Durch das Auftreten unerwarteter Kostensteigerungen und Terminverschiebungen kann eine schlechtere Stimmung in der Planung entstehen, wodurch der Druck auf alle Beteiligten noch mehr ansteigt. Ein weiterer negativer Aspekt ist, dass durch die langen Reaktionszeiten die Transparenz und Nachvollziehbarkeit im Planungsprozess leiden: Stakeholder verlieren das Vertrauen in die Terminprognosen und die Zuverlässigkeit der Bauplanung.

Angepasster Vorgehensprozess zur Baufortschrittskontrolle mit Einbezug eines digitalen Bauwerksmodells (neuer Prozess)

Der moderne Bauprozess beginnt mit der Erstellung eines digitalen Bauwerksmodells. Dieses Modell dient als zentrale Informationsquelle für das gesamte Projekt. Um eine reibungslose Zusammenarbeit zwischen verschiedenen Softwaresystemen zu ermöglichen, wird dieses Modell im offen dokumentierten IFC-Datenformat exportiert. Auf Basis der im Modell hinterlegten Termininformationen wird digital ein Terminplan erstellt und mit den vordefinierten Attributen des IFC-Datenschemas ins Gesamtmodell integriert.

Während der Bauausführung werden die Arbeiten direkt über mobile Endgeräte mit einem Baujournal-App dokumentiert. Dies ermöglicht eine unmittelbare Erfassung des tatsächlichen Baufortschritts vor Ort und verhindert langwierige Nacherfassungsarbeiten. Siehe Flussdiagramm Abbildung 26 und nachfolgende Beschriebe:

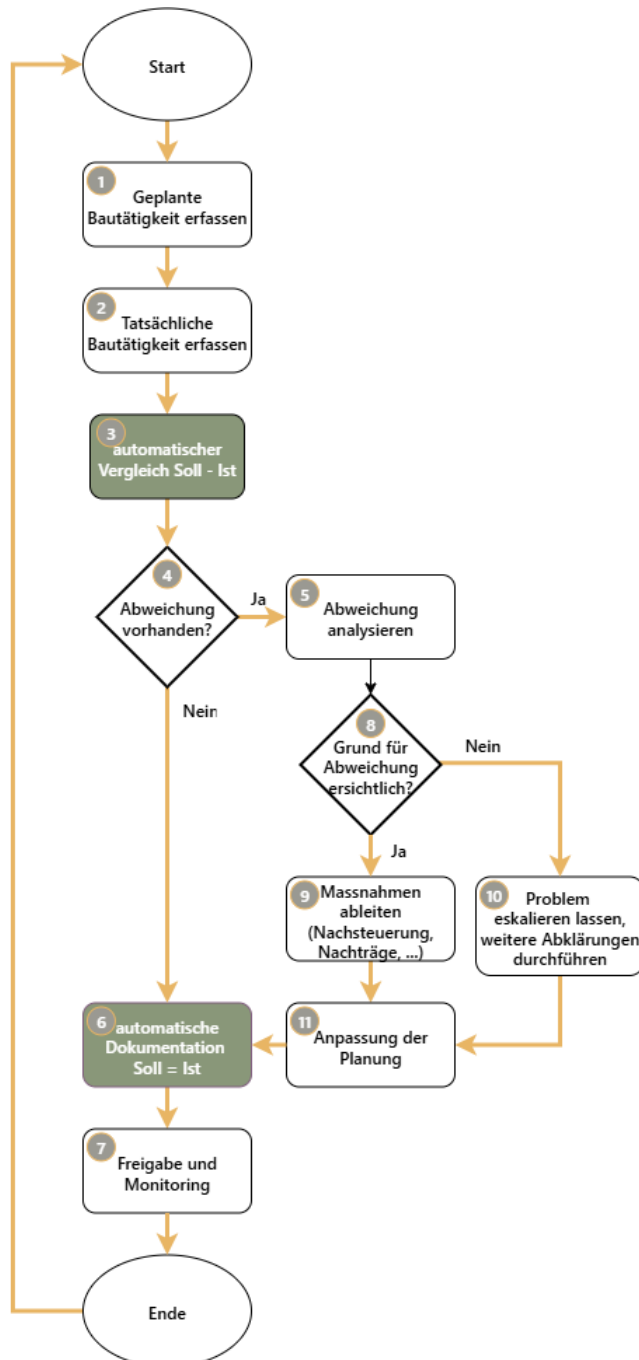


Abbildung 26; Angepasster Vorgehensprozess zur Baufortschrittskontrolle mit Einbezug eines digitalen Bauwerksmodells (morgen, neuer Prozess)

Start

1. **Geplante Bautätigkeit erfassen**
Die Bautätigkeit wird geplant und in einem Terminprogramm erfasst.
2. **Tatsächliche Bautätigkeit erfassen**
Die aktuellen Daten werden auf der Baustelle erfasst.
3. **Automatischer Vergleich Soll – Ist**
Die geplanten und tatsächlichen Bautätigkeiten werden automatisch verglichen.
4. **Abweichung vorhanden?**
Gibt es eine Abweichung zwischen dem Bauprogramm und der Realität? Entscheidung mit Ja oder Nein
5. **Abweichung analysieren**
Die Abweichung wird analysiert und der Grund dafür gesucht.
6. **automatische Dokumentation: Soll = Ist**
Da die verschiedenen Stände automatisch dokumentiert sind und dass Soll dem Ist entspricht, wird der nächste Schritt angegangen.
7. **Freigabe und Monitoring**
Der Prozess geht in die Überwachung über.
8. **Grund für Abweichung ersichtlich?**
Ist der Grund für die Abweichung ersichtlich? Entscheidung mit Ja oder Nein. Beispielsweise weniger Arbeiter auf der Baustelle als geplant.
9. **Massnahmen ableiten**
Konkrete Gegenmassnahmen werden beschlossen (z.B. Nachsteuerung, Nachträge ...).
10. **Problem eskalieren lassen, weitere Abklärungen durchführen.**
Falls der Grund unklar ist, wird das Problem eskaliert und es folgen weitere Abklärungen. Danach Rückführung zu 11 (Anpassung der Planung).
11. **Anpassung der Planung**
Aktualisierung und Korrektur der ursprünglichen Planung. Dies kann das Terminprogramm betreffen aber auch die geometrische Planung.

Ende

Mit dem Ende des Ablaufs beginnt er wieder von vorne bis das Ende der Bauarbeiten erreicht ist.

Anders als im bestehenden Prozess (heute) wird (morgen) der automatische Abgleich mit dem Terminprogramm nun in deutlich kürzeren Intervallen ausgeführt. Dadurch können Abweichungen sehr viel früher erkannt und korrigiert werden, bevor sie sich zu grösseren Problemen entwickeln. Kleine Anpassungen lassen sich zusammen mit der visuellen Unterstützung im digitalen Bauwerksmodell vereinfacht im laufenden Prozess umsetzen. So sind nur noch gezielte Steuerungsmassnahmen notwendig, anstelle von aufwändigen Besprechungen und Diskussionen im grossen Gremium. Dies kann die Geschwindigkeit der Entscheidungsfindung markant erhöhen und die Flexibilität der Bauleitung. Anstatt auf starre Abläufe angewiesen zu sein, kann das Projektteam agil reagieren und den Baufortschritt laufend optimieren. Dadurch wird eine präzisere Steuerung möglich, die Termin- und Kostenrisiken reduziert.

Ein weiterer Vorteil besteht darin, dass die kontinuierlichen Rückmeldeschleifen Transparenz schaffen und die Nachvollziehbarkeit der Entscheidungen verbessern. Durch die regelmässigen Updates haben alle Beteiligten ein aktuelles Bild des Projektstands, was die Kommunikation erleichtert und das Vertrauen in den Prozess stärkt. Zudem sinkt die Wahrscheinlichkeit von Eskalationen, da Probleme bereits in einem frühen Stadium gelöst werden können.

5.1.3.2 Fachlich-konzeptuelle Modellierung

Die fachliche Architektur der Low-Code-App beschreibt den Aufbau und die Organisation der Anwendung aus Sicht der Funktionen und Geschäftslogik, unabhängig von der technischen Umsetzung im Detail. Sie legt fest, welche fachlichen Bestandteile die App hat, wie diese zusammenarbeiten und welche Aufgaben sie erfüllen. Sie beschreibt zudem die visuelle Gestaltung, den einzelnen Seitenaufbau, die Navigation und die umzusetzenden Module.

► OBERFLÄCHE

Die Oberfläche der App ist konsequent auf den Arbeitsalltag der Bauleitung ausgerichtet. Dabei folgt sie einer strukturierten, modularen Anordnung. Die Bedienung der Applikation wird mit dem iPad oder Tablet getätigt.

Der Seitenaufbau gliedert sich in drei Teile:

1. **Kopfzeile:** Projektname
2. **Bearbeitungsfeld** und/oder **Informationen**
3. **Fusszeile:** Export oder Speichermöglichkeiten

Visuelle Gestaltung

Das Low-Code-Frontend der prototypischen Microsoft Power App verfolgt das Ziel, die in Kapitel 2.3.1 beschriebene Information eines Baujournals digital abzubilden und zu manipulieren. Dabei beginnen die Nutzenden ihren Benutzungsfluss beim «Welcome-Bildschirm», wie in der Abbildung 27 ersichtlich.



Der **Welcome-Bildschirm** gibt eine Übersicht über die verfügbaren Projekte. Hier kann sich der Nutzer entscheiden, ob er über die Projektübersicht zu den Bauteilen gelangen will oder zuerst mit der Erstellung des Baujournals starten möchte.

Abbildung 27; eigene Visualisierung vom «Welcome-Bildschirm» in Microsoft PowerApps

Die zentrale Komponente dieser Forschung ist das Low-Code-Frontend «ausgeführte Arbeiten: Übersicht», welche die Informationen aus der IFC-Datei wie in der Abbildung 28 ersichtlich, darstellt. Dabei werden die

importierten Excel-Daten, welche in diesem Prototyp als zentrale Datenbank fungiert, für die Nutzenden mittels REST-API-basierter Verbindung sichtbar und auswählbar gemacht.

Globalld	Geschoss	Raum	Bauteil	Completion	ActualDuration	ScheduleDuration	Länge	Dicke	Höhe
00tMo7QcxqWdlGvc4sMN2A	Erdgeschoss	3.5 Zi Whg	wall	0.8	993938400000		3.05	0.2	3.5
1EnkEEOj90gLyismkc_TU	1Obergeschoss	4.5 Zi Whg	wall	0.1	45.5		1	0.5	3.5
2ATDWCx5XE2AzvVLDYG8bA	3Obergeschoss	5.5 Zi Whg	wall	0.99	7.5	1.75	0.67	0.15	3.5
0jHn52Hxj6ReQJZZpi3Xtu	3Obergeschoss	Wohnen	wall	0.5	9.3			0.12	2
256QK5oVDDVPXeMfiz4xbQ	Dachgeschoss	Reduit_Keller	wall	0.25	0.35			0.3	3.5
	Untergeschoss	Treppenhaus	wall	1				0.4	3.5
	Untergeschoss	Technik	wall	1				0.2	3.5
	Dachgeschoss	2.5 Zi Whg	wall	0				0.2	3.5
				#DIV/0!					
				#DIV/0!					
				0		1.75			
				#DIV/0!					
				#DIV/0!					
				#DIV/0!					

Abbildung 28; eigene Visualisierung von «ausgeführte Arbeiten: Übersicht» Variante 1 in Microsoft PowerApps

Auf der Ansicht **ausgeführte Arbeiten: Übersicht** können Bauteile aus der IFC-Datei anhand von Informationen geprüft werden. Die Bauteile weisen Informationen zum: Geschoss, Zugehörigem Raum, Bauteil, sowie *Completion* (Fertigstellungsgrad), *ActualDuration* (Tatsächlich gebrauchte Dauer), *ScheduleDuration* (geplante Dauer) und der Länge, Dicke und Höhe, aus.

Möchten die Nutzenden ein Bauteil bearbeiten, wird dieses in der Liste zuerst angeklickt und über die Schaltfläche «**Status Bearbeiten**» gelangt man zur nächsten Seite für die Manipulation.

Über die Schaltfläche «**Baujournal erstellen**» kann für das entsprechend ausgewählte Bauteil ein entsprechendes Dokument erstellt werden, was jedoch auf später beschriebenen Ansichten erfolgt.

Abbildung 29; eigene Visualisierung von «Status bearbeiten» in Microsoft PowerApps

Die Ansichtseite **Status bearbeiten** dient dazu die Bauteile auf der Baustelle zu bearbeiten.

Das Bauteil wird ausgewählt und über den Knopf «**Edit**» bearbeitet. Dies betrifft ausschliessliche die Attribute *ActualDuration* (Tatsächlich gebrauchte Dauer), und *Completion* (Fertigstellungsgrad). Mit dem Kopfdruk auf «**Submit**» werden die neuen Werte in der Datenbank gespeichert.

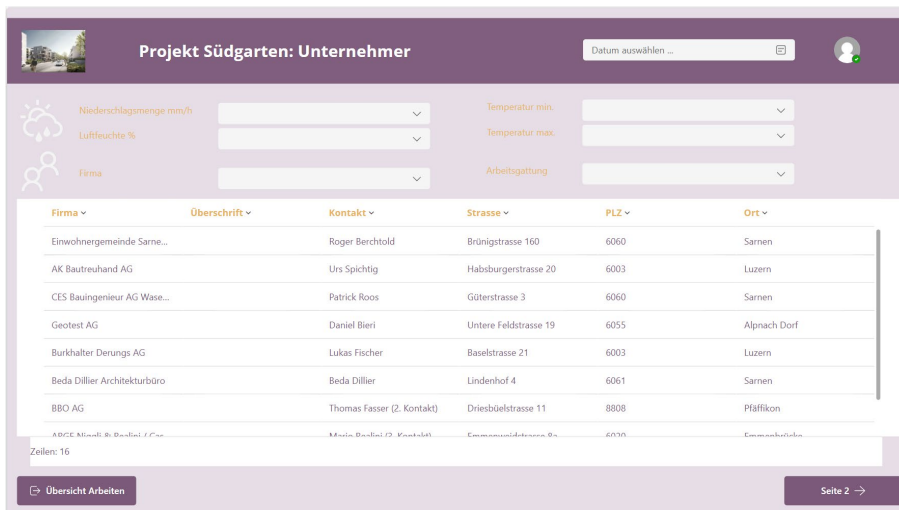


Abbildung 30; Visualisierung von Seite 1 «Unternehmerangaben» in Microsoft PowerApps

Die Unternehmerangaben-Seite 1 gehört zu den geforderten Informationen für die Erstellung eines Baujournals nach SIA112.

Hier wird im oberen Teil der Seite das *Datum*, die *Niederschlagsmenge* [mm/h], *Luftfeuchte* [%], *Temperatur min*, *Temperatur max*, angegeben. Darunter wird die entsprechende *Firma* mit der geleisteten *Arbeitsgattung*, sowie den Arbeitenden definiert.

In der Fusszeile gelangt man über den Knopf «Übersicht Arbeiten» zur Übersicht zurück oder über die Schaltfläche «Seite 2» zur nächsten Seite des Baujournals.

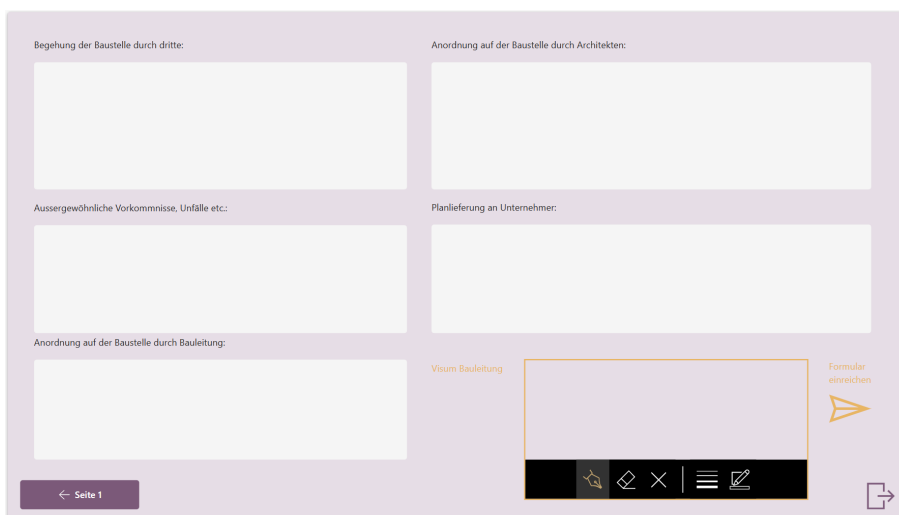


Abbildung 31; Visualisierung von Seite 2 «Unternehmerangaben» in Microsoft PowerApps

Auf dieser Seite 2 der Unternehmerangaben werden zusätzliche Informationen festgehalten, wie etwa: die Begehung der Baustelle durch dritte, Aussergewöhnliche Vorkommnisse, Unfälle etc., Anordnung auf der Baustelle durch Architekten, Planlieferung an Unternehmer, Anordnung auf der Baustelle durch Bauleitung, Anordnung auf der Baustelle durch Architekten, Planlieferung an Unternehmer.

Ganz unten kann das Baujournal **visiert** und mittel «Formular einreichen» an die Bauleitenden übermitteln. Man kann hier über das «Exit-Symbol» aber ebenfalls wieder zu den ausgeführten Arbeiten zurück gelangen.

Viewer-Erweiterung für die visuelle Unterstützung der Bauteilauswahl

Zur Orientierung im digitalen Bauwerksmodells helfen Strukturierungen wie Räume, Achsen, Bauetappen, usw. Meistens hilft die Visualisierung mit einem Viewer für zusätzliche Sicherheit bei der Verortung, weshalb zwei Prototypen der Applikation erstellt wurden. Die Variante 2, wie in Abbildung 32 ersichtlich, Integriert den IFC-Viewer als Erweiterung der ausgeführten Arbeiten. In Microsoft Power Apps wird dazu eine *PowerBI-Kachel* via *Tracer* eingebettet, welche eine Darstellung aus einem Dashboard nach Applikationsprogrammierer:in-Wahl visualisiert. Alle anderen Funktionalitäten sind identisch wie in der vorangehend beschriebenen Ansicht **ausgeführte Arbeiten: Übersicht**.

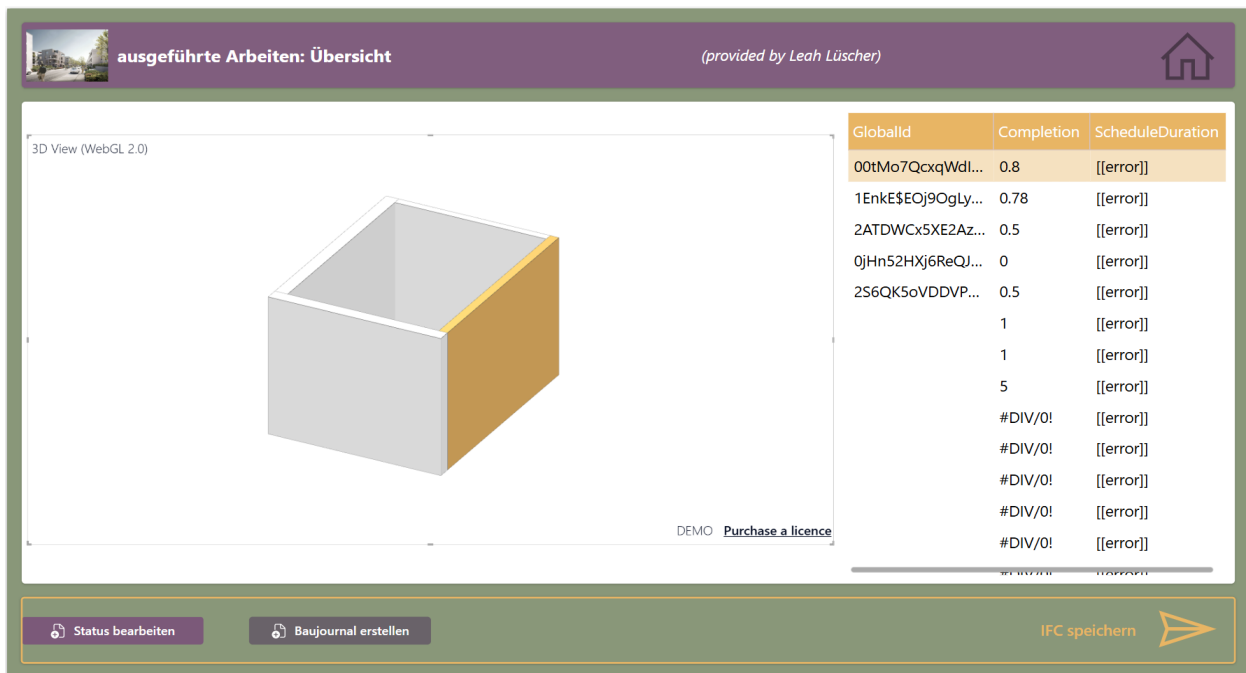


Abbildung 32; eigene Visualisierung von «ausgeführte Arbeiten: Übersicht» Variante 2 verknüpft mit dem Power BI Viewer via Tracer in Microsoft PowerApps

► LAYOUT (Module und Navigation)

Das Layout der Applikation wird durch die Anordnung und Struktur der Module sowie der Navigationsflüsse definiert. Nachfolgend werden diese näher beschrieben, ergänzend ist in der Abbildung 33 die Benutzerführung ersichtlich.

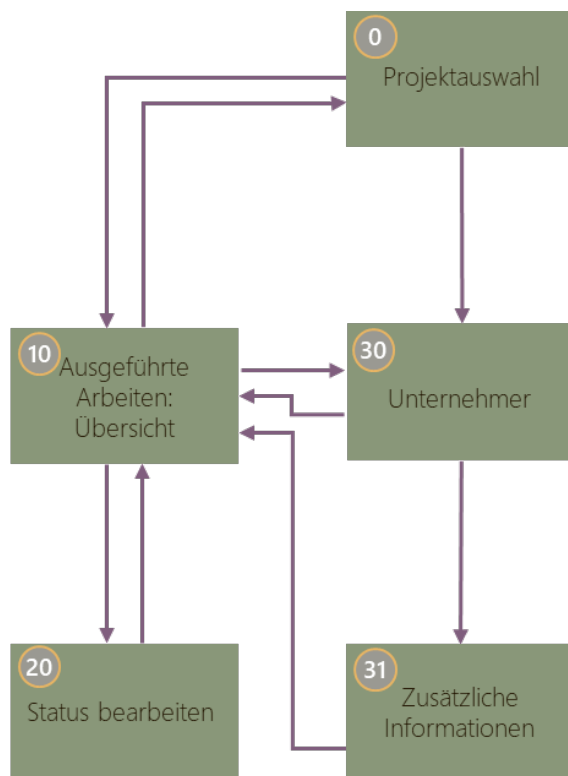


Abbildung 33; Benutzerführung in der Applikation (eigene Abbildung)

0 Beim Starten der App erreichen die Nutzenden den Welcome-Bildschirm, welcher ihm eine Projektauswahl zeigt. Hat diese sich für ein Projekt entschieden und ausgewählt, können sich die Nutzenden einerseits für die *Projektübersicht* oder fürs *Baujournal erstellen* entscheiden.

10 Entscheiden sich die Nutzenden für die Projektübersicht, könne sie Bauteile aus dem IFC in einer Tabelle betrachten. Über einen Knopfdruck lassen sich die Bauteile auswählen und leiten den Nutzer zur nächsten Seite für die Weiterverarbeitung.

10 (Variante) Zur Übersichtlichkeit des gewählten Bauteils oder der visuellen Unterstützung kann auch eine Visualisierung von einem PowerBI-Dashboard eingebettet werden.

20 Auf dieser Seite können die Bauteile bearbeitet und im Anschluss in der Datenbank (Excel) gespeichert werden. Über einen Knopf zurück gelangen die Nutzenden wieder zur Übersicht der ausgeführten Arbeiten und könne sich nun dazu entscheiden ein Baujournal zu erstellen.

30 Auf der folgenden Seite können die Nutzenden *Datum*, *Wetter* und *Unternehmer* mit der dazu gehörigen *Arbeitsgattung* und den verfassenden *Mitarbeiter* auswählen.

► STRUKTUR und Schnittstellen

Technisch muss das Frontend die modulare Anordnung der vier Hauptbereiche: Projektdaten, Personal, Tätigkeiten und Zusatzfelder/Visum auf Tablets unterstützen und eine nachgelagerte Synchronisation mit einer zentralen Datenbank ermöglichen. Zudem sind Schnittstellen für den IFC-Import und -Export über CSV- oder REST-API sowie Anbindungen an Wetter- und Unternehmensverzeichnisdienste erforderlich. Ergänzend sollen Exportfunktionen in Power BI oder Excel bereitgestellt sein, um vertiefte Soll-Ist-Analysen zu ermöglichen.

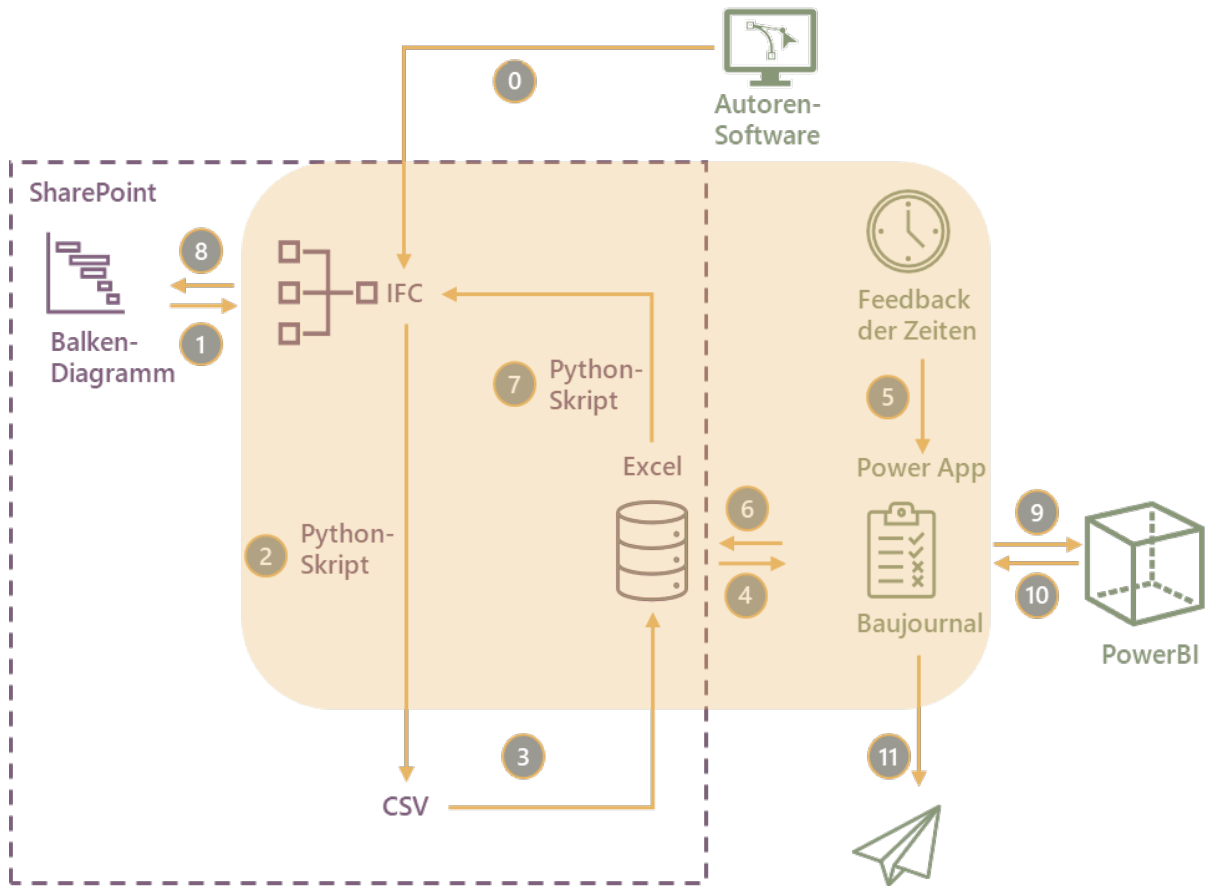


Abbildung 35; Datenfluss in der Applikation (eigene Abbildung)

Folgend werden die Datenflüsse aus der Abbildung 35 erklärt. Der orange eingefärbte Bereich stellt dabei den Umfang des Prototyps dar:

0 Speicherung IFC

Der gesamte Datenfluss beginnt mit dem Speichern des Referenzmodells als IFC-Datei in einer beliebigen Autorensoftware.

1 Speicherung Terminplanung

Als Ausgangspunkt dient das **Balken-Diagramm** der Bauablaufplanung und die IFC-Datei. Die Bauleitung überträgt in der IFC-Datei nun die Attribute *ScheduleStart* und *ScheduleFinish* bei den bauteilspezifischen Entitäten *IfcTaskTime*. Wobei *ScheduleStart* jeweils den geplanten Arbeitsbeginn und *ScheduleFinish* das geplante Arbeitsende vom Bauteil widerspiegelt.

2 Extraktion von Daten aus dem IFC in ein CSV

Über ein **Python-Skript** werden diese Informationen von der IFC-Datei in eine CSV-Datei überführt, dieses kann im Anhang A2 und Anhang D eingesehen werden. Das Skript folgt dabei nachfolgendem Programmablauf:

Zu **Beginn des Skripts** werden verschiedene **externe Bibliotheken** eingebunden, die für das Auslesen und Verarbeiten von IFC-Daten essenziell sind. Die Python-Module: *pathlib*, *ifcopenshell*, *ifcopenshell.geom*, *pandas* und *math* bieten Funktionalitäten zur Pfadangabe, zum Zugriff auf das IFC-Modell, zur geometrischen Analyse der Bauteile sowie zur Datenmanipulation und -ausgabe in Tabellenform. Diese Bibliotheken bilden somit das technische Fundament des Programms.

Als **zweiter** Schritt werden die **Dateipfade** definiert. Der erste Pfad verweist auf die Quelldatei im IFC-Datenformat, welcher primär die GUID und Geometriedaten eines Bauwerks enthält. Der zweite Pfad legt den Speicherort der erzeugten CSV-Datei fest.

Der **dritte** Arbeitsschritt ist die Definition der **Hilfsfunktionen** für den IFC-Datenzugriff über *ifcopenshell*. Dabei unterstützen drei Funktionen die Datenextraktion. *get_container()* identifiziert die Geschossebene *ifcBuildingStorey* auf der sich die Elemente befinden. *get_psets()* extrahiert die Property Sets aus einem Element und wandelt sie in ein Dictionary um. Dadurch werden beispielsweise Masse wie Länge oder Dicke verfügbar gemacht. *get_bbox_dims()* erzeugt eine geometrische Form des Elements und berechnet daraus die räumliche Ausdehnung (Länge, Breite, Höhe). Diese Methode wird nur angewendet, wenn keine Informationen aus den bauteilspezifischen PropertySets vorliegen.

Im nächsten und **vierten** Teil ist die **Hauptlogik** und somit die Datenextraktion und -strukturierung beschrieben. Die Hauptfunktion *export_ifc()* bildet das Herzstück des Skripts. Sie öffnet die IFC-Datei und initialisiert die Geometrieeinstellungen. Anschließend filtert sie die Bauteiltypen beispielsweise *IfcSpace*, *IfcWall* und *IfcWallStandardCase* (Räume und Wände) heraus, um aus diesen Attributen die relevanten Informationen zu extrahieren.

Für jedes Element werden folgende Spalten (je Attribut) für die Informationsablage definiert:

- GlobalId (GUID oder Globally Unique Identifier)
- Bauteiltyp
- Geschoss
- Raumname
- Länge
- Höhe
- Dicke

Ergänzt werden folgende Attribute als Speicherort für die später benötigten Werte aus der Bauablaufplanung:

- DurationType
- ScheduleDuration
- ScheduleStart
- ScheduleFinish
- Early Start
- EarlyFinisch
- LateStart
- LateFinish
- FreeFloat
- TotalFloat
- IsCritical
- StatusTime
- ActualDuration
- ActualStart
- ActualFinisch
- RemainingTime
- Completion

Als **letzter Schritt des Skripts** wird die **CSV-Datei erstellt und exportiert**. Alle erfassten Informationen werden in *pandas-DataFrame* überführt, welches die strukturierte Grundlage für die Ausgabe bildet. Die Reihenfolge der Spalten ist vorgegeben und deckt sowohl geometrische als auch terminologische Attribute ab. Zum Schluss wird die Tabelle als CSV-Datei mit Semikolon als Trennzeichen exportiert und eine Erfolgsmeldung wird auf der Konsole ausgegeben.

In der letzten Zeile wird geprüft, ob das Skript direkt ausgeführt wird. Falls ja, wird die Hauptfunktion *export_ifc()* aufgerufen. Dadurch wird sichergestellt, dass der Exportprozess nur dann startet, wenn das Skript nicht als Modul importiert, sondern eigenständig ausgeführt wird.

3 Formatieren einer CSV-Datei als Excel

Die **CSV-Datei** wird manuell im Excel als Datenimport geöffnet, als Tabelle formatiert und als Excel-Datei gespeichert. Damit liegen die Daten in tabellarischer Form vor und können einfach weiterverarbeitet werden.

4 Datenbank in PowerApps

Die **Excel-Tabelle** dient hier als Datenbank und ist für alle Nutzenden der Applikation zugänglich und sichtbar. Sie wird nach den Bedürfnissen in der Microsoft PowerApps eingebettet. Darin werden alle Informationen mit den in Relation stehenden Bauteilen mitgegeben. Beispielsweise die GUID und die Bauteilart, sowie die örtliche Lokalität im digitalen Bauwerksmodell.

Jedes funktionale Modul greift in Echtzeit auf den lokalen Datenspeicher der Applikation zu, um Daten zu lesen und zu schreiben. Sobald eine Netzwerkverbindung besteht, werden die Daten automatisch mit dem zentralen SharePoint synchronisiert.

5 Fortschritts-Aufnahme auf der Baustelle

Die Bauleitung nimmt beim Baustellenrundgang das **Feedback der Zeiten** sowie die aktuellen Fertigstellungsgrade der Bauteile auf. Dazu nutzt sie die Eingabemaske auf der Ansichtseite «Status bearbeiten» der Applikation.

6 Rückführung der Baufortschritts-Informationen in die Datenbank

Nach Abschluss der Eingaben exportiert die App die veränderten Datensätze erneut in die Excel-Datei. Ab diesem Zeitpunkt stehen allen Nutzenden wieder die aktuellsten Termin- und Baujournalinformationen zur Weiterverarbeitung zur Verfügung.

7. Speicherung der Daten von der Datenbank in die IFC-Datei

Mit dem zweiten **Python-Skript** werden diese Informationen von der Excel-Datei zurück in die IFC-Datei übertragen. Das Skript liest dazu zuerst die Attribute der Excel Datei und schreibt die neuen Fertigstellungsgrade *Completion* der Bauteile in die IFC-Datei. Die berechnete *RemainingTime*, also die noch verbleibenden Tage bis zur geplanten Fertigstellung, werden ebenfalls ins IFC-Modell übertragen. Das Skript kann im Anhang A4 und im Anhang E eingesehen werden und folgt dabei nachfolgendem Programmablauf:

Zu **Beginn** bindet das Skript mehrere Python-Bibliotheken ein, die für die Datenverarbeitung und -erzeugung erforderlich sind. *Pandas* dient dem Zugriff auf und der Verarbeitung tabellarischer Daten aus Excel. *ifcopenshell* wird zum Erstellen, Bearbeiten und Speichern von IFC-Bauwerksmodellen verwendet. Das Modul *guid* zur Generierung eindeutiger Identifikationsnummern für IFC-Entitäten sowie *datetime* zur Erzeugung von Zeitstempeln für Aufgaben- und Bauzeitinformationen.

Im **zweiten** Schritt werden die Dateipfade definiert: Excel-Dateipfad, IFC-Eingabepfad, IFC-Ausgabepfad.

Als **drittes** werden die Hilfsfunktionen definiert. Mehrere nachfolgend beschriebene Funktionen kapseln dabei wiederkehrende Aufgaben:

Generate_guid() erstellt eine eindeutige, komprimierte IFC-GUID.

create_property_single_value() erzeugt eine IFC-Eigenschaft mit einem einzelnen Wert (zum Beispiel Höhe oder Länge).

CreatePropertySet() fasst mehrere Eigenschaften zu einem *PropertySet* zusammen.

Mit *create_quantity_length()* wird eine IFC-Mengenangabe für Längenwerte angelegt.

Diese Hilfsfunktionen vereinfachen die Erstellung standardkonformer IFC-Entitäten und sorgen für einheitliche GUID- und Eigentumsinformationen.

Als nächster und **vierter** Schritt wird die Hauptlogik definiert. Die Hauptlogik durchläuft jede Zeile der Excel-Datei und überträgt die darin hinterlegten Attributinformatoren in die IFC-Struktur.

Falls sie im Modell noch nicht vorhanden sind, werden Geschosse *IfcBuildingStorey* angelegt und mit dem Projekt verknüpft, also neu Instanziiert.

Die Raumentitäten *IfcSpace* werden mithilfe der in Excel angegebenen GUID erstellt und in das jeweilige Geschoss eingefügt.

Entitäten der Wände *IfcWall* werden erzeugt und ebenfalls dem jeweiligen Geschoss zugeordnet.

Eigenschaften für Höhe und Länge werden als *IfcPropertySingleValue* angelegt, in einem vordefinierten Eigenschaftenset *Pset_WallCommon* gebündelt und mit der Wand verknüpft.

Zeitinformationen *IfcTaskTime* werden für jede Zeile erstellt und mit Start- und Endzeiten sowie weiteren Terminattributen versehen.

Anschliessend als **fünfter** Schritt wird die mit neuen Informationen angereicherte IFC-Datei exportiert. Die neu erstellten oder ergänzten IFC-Entitäten werden ins geöffnete Bauwerksmodell eingefügt und anschliessend unter dem angegebenen Ausgabepfad gespeichert.

Zum Schluss wird mit einer Konsolenmeldung der erfolgreiche Export bestätigt.

8 Abgleich Aktuelle Termine aus dem IFC mit Bauplanung

Die neuen beziehungsweise aktualisierten Daten fließen wieder zurück in die Bauplanung, wo nach Bedarf entsprechende Steuerungsmassnahmen eingeleitet werden können.

9 Darstellung Termin-Daten in PowerBI

Parallel dazu erstellt ein automatisierter Workflow in Microsoft Power Automate eine aktualisierte und visuell aufbereitete Bauplanungsübersicht in Microsoft Power BI.

10 Darstellung 3D-Viewer in PowerApps

Die Darstellung des digitalen Bauwerksmodelles, den so genannten Viewer wird über eine direkte Integration eines Microsoft-PowerBI-Dashboards gelöst. Dieser visualisiert die Daten des gewählten Dashboards mithilfe der zusätzlichen Applikation von Tracer in Microsoft PowerBI.

11 Export Baujournal als PDF

Durch die Speicherung der neuen Daten im Zusammenhang mit den erweiterten Informationen auf dem Baustellenrundgang wird den betroffenen Nutzenden ein PDF des Baujournals erstellt und abgelegt.

Einfluss auf Schnittstellen

Durch diesen klar definierten Datenkanal IFC → CSV → Dataverse/PowerApps → IFC wird sichergestellt, dass alle Planungs-, Modell- und Journaldaten konsistent bleiben und in Echtzeit oder nach Bedarf synchronisiert werden. Gleichzeitig erlaubt die Architektur, neue Schnittstellen beispielsweise zu Wetterdiensten oder externen BI-Tools) unkompliziert hinzuzufügen, ohne den Kernprozess zu unterbrechen. Allerdings sind die Schnittstellen anfällig und fallen ändernden Datenschutzrichtlinien zum Opfer.

5.1.4 Diskussionsergebnisse der Fokusgruppe in Bezug auf den Prototypen

Im Folgenden werden die kondensierten Ergebnisse vom Expertengespräch (Fokusgruppe) mit der Praxispartnerin dargelegt. Diese sind thematisch in EXPERTENAUSSAGEN und NUTZWERTANALYSE unterteilt. Die Aussagen geben einerseits spezifische Antworten auf die Fragestellungen dieser Forschung. Mit der Analyse werden andererseits qualitative Kriterien transparent gewichtet und im gleichzeitig quantitativ beurteilt.

► EXPERTENAUSSAGEN

An der Diskussion der Fokusgruppe nahmen acht Nutzende aus der gleichen Unternehmung teil, siehe Tabelle 5. Die Durchführung der Fokusgruppe fand physisch vor Ort statt, wobei zwei Personen virtuell zugeschaltet waren. Da alle hauptsächlich mit Microsoft PowerApps im Zusammenhang mit dieser Technologie arbeiten, könnte die Varianz der Aussagen begrenzt sein.

Tabelle 5: Teilnehmende der Fokusgruppe, teilanonymisiert über eine generische ID

ID	Funktion	Unternehmen
1	Microsoft Power Solutions Experte	Praxispartner
2	Head BIM Civil Engineering, Regionalebene	Praxispartner
3	Head Lean Construction Civil Engineering, Regionalebene	Praxispartner
4	Digital Technology Manager, Divisionsebene	Praxispartner
5	Operational Excellence / Prozessoptimierung, Regionalebene	Praxispartner
6	BIM Specialist / Information Management, Regionalebene	Praxispartner
7	Head of BIM Civil Engineering, Divisionsebene	Praxispartner
8	Head of Lean Civil Engineering, Divisionsebene	Praxispartner

Im Hauptteil wurden folgende Fragen diskutiert, woraus die folgenden Erkenntnisse beschrieben wurden:

- **Chancen:**
Was sind die Chancen im Vergleich zu herkömmlichen Programmiermethoden?
- **Herausforderungen und mögliche Risiken:**
Wo könnte Low Code an seine Grenzen stossen und welche Risiken gilt es zu beachten?
- **Typische Arbeitsabläufe im Bauwesen:**
Wie laufen Prozesse derzeit in eurem Alltag ab?
An welchen Stellen könnte Low Code sinnvoll eingebunden werden, um Abläufe zu optimieren?
- **Technische Voraussetzungen:**
Welche Infrastruktur oder Systemanbindungen sind notwendig, damit eine Low-Code-Plattform reibungslos funktioniert?
Gibt es spezielle Anforderungen an Sicherheit oder Schnittstellen?
- **Auswahl geeigneter Low-Code-Plattformen:**
Mit welchen Tools habt ihr bereits Erfahrungen und welche würdet ihr für unser Szenario empfehlen?

Chancen und Herausforderungen

Low-Code-Plattformen wie die Microsoft Power Platform bieten viele Möglichkeiten, besonders im internen Einsatz (1). Hier lassen sich Prozesse oft schnell und effizient digitalisieren und auch die Skalierbarkeit ist gut handhabbar (4). Doch sobald externe Partner ins Spiel kommen, ändern sich die Spielregeln. Themen wie Datensicherheit, Zugriffsrechte und Freigaben werden zu entscheidenden Stolpersteinen (5). Dabei kommen in der Praxis Fragen auf wie:

- Wer darf welche Daten wo und wie abrufen?
- Welche Sicherheitsmechanismen müssen eingebaut werden (1)?

Gerade die Freigabe an externe Nutzer sorgt immer wieder für Probleme (8) und erfordert oft individuelle Plattformlösungen (5).

Grundsätzlich lässt sich mit Low-Code nahezu alles umsetzen (1). Die Fragen sind nur:

- Wie hoch ist der Aufwand?
- Ergeben die Massnahmen strategisch Sinn?
- Und wie stark hängt man dabei vom Wohlwollen des Plattformanbieters ab (1)?

Für kleine, spezifische Anwendungsfälle (4) oder einfache SharePoint-Listen (3) eignet sich Low-Code hervorragend. Doch je näher man an komplexe Enterprise-Applikationen heranrückt, desto deutlicher treten die Grenzen zutage (3). Die Applikationslandschaft vieler Unternehmen ist schlicht nicht für Low-Code konzipiert (4). Hinzu kommt, dass viele Anwender nur über begrenzte technische Kenntnisse verfügen, was die Integration in eine Enterprise-Architektur erschwert (4).

Oft bleibt eine Anwendung im Prototypenstatus hängen (7). Schon bei der nächsten Anforderung ändert sich wieder etwas (7), sodass eine nachhaltige, skalierbare Umsetzung in die Ferne rückt. Wer eine langfristig tragfähige Lösung anstrebt, muss die Entwicklung irgendwann von PowerApps auf andere Technologien umstellen (3), was zusätzliche Ressourcen und Know-how erfordert. Entscheidend ist daher eine klare Governance-Struktur (3) und eine Organisation im Hintergrund, die gebaute Anwendungen auch dauerhaft unterstützen kann (3).

Nicht zu unterschätzen ist zudem der Faktor Verständlichkeit: Der ursprüngliche Ersteller kennt Aufbau und Funktionsweise seiner App im Detail, während es für andere oft schwer ist, den Faden nachzuvollziehen (3). Das wirft die Frage auf, ob Branchen wie das Bauwesen Applikationen wollen, die sie selbst kaum verstehen (5). Manchmal erscheint es sinnvoller, auf spezialisierte Lösungen wie ACC zu setzen (5), bei denen man sicher sein kann, dass sie professionell betrieben werden.

Letztlich gilt: Low-Code entfaltet seine Stärken vor allem intern (1), in klar begrenzten Szenarien (4) und dann, wenn die Anwendung für den eigenen Bedarf gedacht ist (1). Für den externen Einsatz ist Low-Code nur bis zu einem gewissen Punkt praktikabel (1) und immer nur dann, wenn Aufwand, Sicherheit und Nutzen in einem ausgewogenen Verhältnis stehen. Low-Code eröffnet neue Möglichkeiten, weil im Prinzip jede Person ihre eigenen Use Cases umsetzen kann, vorausgesetzt, die Lösung ist einfach genug und bleibt im überschaubaren Rahmen (4). Dieser Ansatz senkt die Einstiegshürden, denn Low-Code ist für alle gedacht (4). Dadurch können auch Fachabteilungen ohne tiefes IT-Wissen ihre Ideen in funktionsfähige Anwendungen umsetzen.

Ein besonderer Vorteil ist die Geschwindigkeit. Innerhalb eines halben Tages lässt sich mit Low-Code ein Minimum Viable Product erstellen (3), das sofort getestet und im realen Betrieb erprobt werden kann. Anwendungen wie digitale Begehungen mit automatischem Mailversand, wie sie etwa bei Wincasa genutzt werden, sind damit gut umsetzbar. Gerade bei unternehmensnahen Szenarien wie Bautagebüchern, die täglich genutzt werden (3), bietet Low-Code eine schnelle und flexible Alternative zu langwierigen Entwicklungsprojekten.

Die iterative Vorgehensweise wird deutlich erleichtert: Man kann viel schneller zu den Nutzer:innen gehen, diese früh einbeziehen und gemeinsam zu einer Lösung gelangen (7). Anstatt lange auf Feedback zu warten, entsteht ein direkter, dynamischer Austausch, der zu passgenaueren Ergebnissen führt. So entwickeln viele Anwender ihre eigenen Automatismen (7), um alltägliche Aufgaben effizienter zu gestalten.

Skalierbarkeit auf andere Projekte

Bei der Übertragung von Applikationen auf andere Projekte ergeben sich spezifische Herausforderungen. So erfordert die Wiederverwendung häufig zusätzliche Infrastruktur, wie beispielsweise relationale Datenbanken oder SQL, was wiederum mit Lizenzkosten verbunden ist (1). Auch die laufenden Kosten pro Nutzer, beispielsweise 5 Franken pro User, können sich bei einer hohen Nutzerzahl schnell zu beträchtlichen Summen addieren (1). In manchen Fällen lohnt es sich daher, auf Dataverse, die relationale Datenbank von Microsoft, zurückzugreifen. Das Frontend wird in solchen Szenarien oft als Webanwendung realisiert (1).

In der Praxis zeigt sich, dass es im Unternehmen eine Vielzahl von Anwendungen gibt, die für unterschiedliche Projekte eingesetzt werden, darunter etwa 123 Apps und 127 Flows. Nach Projektabschluss besteht allerdings die Einschränkung, dass nur noch der jeweilige Projekteigentümer die Anwendung bearbeiten kann (5). Gerade bei Projekten wie Bauprotokollen, die nach Abschluss als PDF archiviert werden, limitiert dies die Weiterentwicklung und die Anpassung an neue Anforderungen.

Anforderungen und Bedürfnisse der Nutzenden

Für Nutzer einer Anwendung ergeben sich oft Hürden in Bezug auf Sichtbarkeit und Informationsmenge. So kann es beispielsweise vorkommen, dass Anwender die zugrunde liegenden Daten gar nicht einsehen können (5). Auch die Darstellung und die Menge der Informationen innerhalb einer App sind nicht immer optimal abgestimmt oder filterbar, was die Bedienbarkeit erschwert (5).

Um die Übersichtlichkeit zu verbessern, bietet sich beispielsweise eine Filterung nach Gewerk an (2). Ebenso kann eine Filterung nach Arbeitsbereich oder Bauetappe helfen, die relevanten Daten schnell zu finden und die Nutzererfahrung zu verbessern.

Umgang mit dem bauwerksmodellbasierten Baufortschritt

Beim Erfassen des Baufortschritts wird heute oft direkt im Modell gearbeitet. Sobald ein Bauteil fertiggestellt ist, wird es als erledigt klassifiziert (7). Doch genau hier liegt eine Herausforderung, denn wann gilt ein Bauteil wirklich als fertig? In manchen Fällen sogar dann, wenn es zurückgebaut wurde. Das erschwert die Unterscheidung zwischen «ausgeführt» und «nicht ausgeführt». Für eine funktionale Anwendung spielt die Visualisierung eine zentrale Rolle (8). Technisch wird dies häufig über Power Apps Component Framework (PCF)-Komponenten realisiert, die anhand einer GUID die entsprechenden Elemente im Modell identifizieren und darstellen (7).

Auch die Abbildung der verschiedenen Phasen innerhalb eines BIM-Prozesses ist ein wichtiger Bestandteil (7). Hinzu kommt, dass je nach Bauteil unterschiedliche Arbeitsgattungen beteiligt sind (7), was die Standardisierung

erschwert. Eine weitere Herausforderung ist die klare Definition dessen, wann ein Bauteil als fertiggestellt gilt (7), da dies je nach Projektsituation variieren kann.

IFC und Low-Code im App

Die Integration von IFC-Daten in eine Low-Code-App stellt in der Praxis oft projektspezifische Anforderungen. Besonders wichtig ist dabei die Verbindung von Detaillierungsgrad und Terminprogramm (7). Ein gängiger Ansatz besteht darin, den Terminplan und die IFC-Daten über einen gemeinsamen Schlüssel zu verknüpfen. Allerdings liegen diese Informationen meist in separaten Datenbanken vor (7).

► NUTZWERTANALYSE

Die folgende Nutzwertanalyse bewertet den Einsatz einer Low-Code-App für das Unternehmen anhand wichtiger Kriterien gemäss den Rechercheergebnissen wie: Personalressourcen, Datenqualität, Zeitersparnis und Anwendbarkeit im Baustellenkontext. Dabei werden sowohl die Stärken als auch die Einschränkungen dieser Technologie beleuchtet. Die Ergebnisse sind in Abbildung 36 grafisch dargestellt und im Folgenden beschrieben.

Die Teilnehmenden konnten pro Beurteilungskriterium je Gewichtung oder Bewertung eine Einschätzung von «1» bis «4» machen. Ziel dieser Wertebereichswahl war es dabei minimal ein gewichtetes Resultat zu erhalten, da keine zentrische Einschätzung möglich ist.

Aus der Befragung der Fokusgruppe stellte sich heraus, dass die beiden Beurteilungskriterien:

- *Personalressourcen*
- *Fehlerrate / Verringerung der manuellen Nacharbeit / Sicherstellung einer hohen Datenqualität*

von den Gewichtungen her am höchsten eingestuft werden. Gefolgt von der *Softwarewahl* folgen alle anderen Aspekte etwa im selben Rahmen aus. Einzig die Kriteriengruppe *Return on Investment (ROI) und Wiederverwendung* wird mit einer untergeordneten Gewichtung eingestuft.

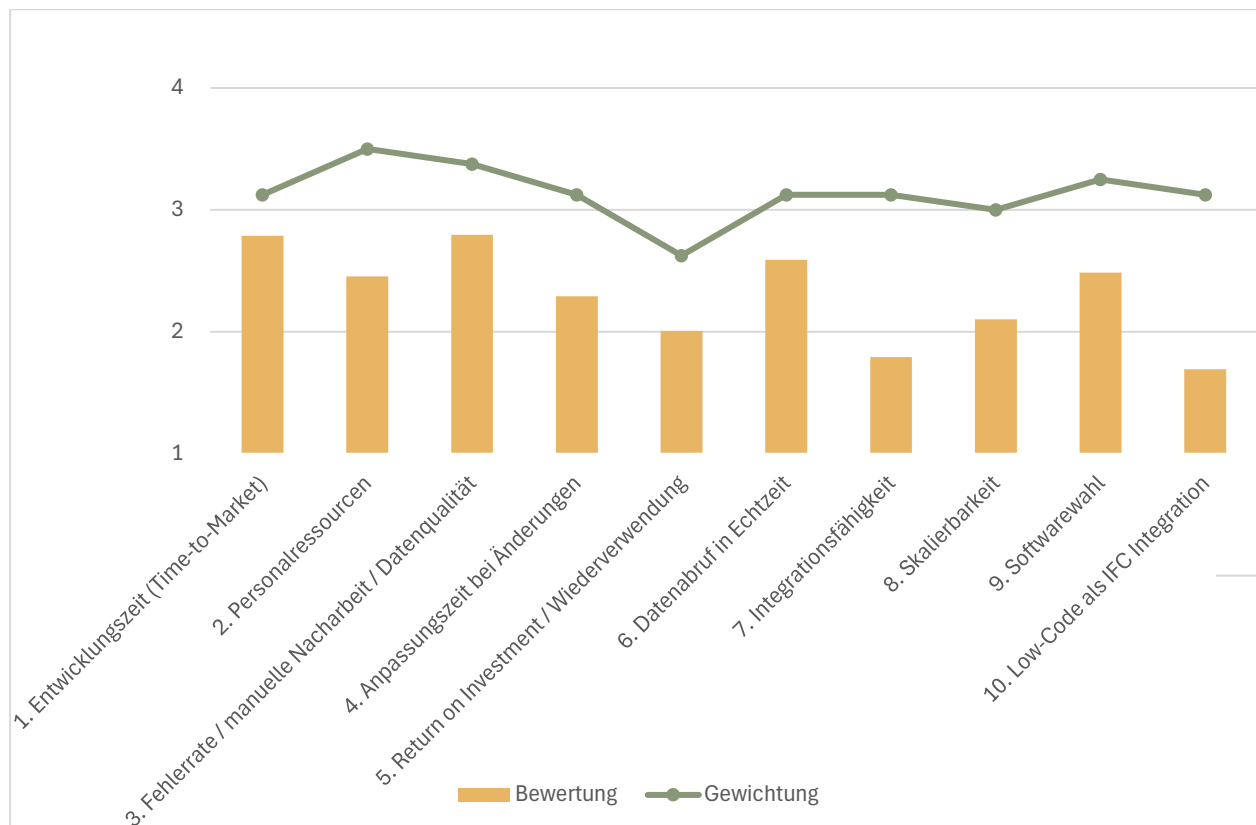


Abbildung 36; Auswertung Nutzwertanalyse (eigene Abbildung)

Bei der Betrachtung der effektiven Bewertung ist ersichtlich, dass für das Unternehmen zwei Faktoren von grösserer Bedeutung sind. Zum einen die Entwicklungszeit (Time to Market) zum anderen die Sicherstellung einer hohen Datenqualität und die Verringerung der manuellen Nacharbeit.

In Projekten ist die Verfügbarkeit auf verlässliche Informationen oft entscheidend dafür, ob Prozesse reibungslos

verlaufen oder Verzögerungen entstehen. In einem solchen Fall ist weder die Terminplanabfrage noch die Rückmeldung erfasster Arbeitszeiten auf ein entsprechendes Element möglich. Auch der praktische Nutzen auf der Baustelle selbst ist nicht zu unterschätzen. Anwendungen, die mit Low-Code entwickelt wurden, können Prozesse vor Ort unterstützen, vereinfachen und beschleunigen, indem sie relevante Daten in Echtzeit bereitstellen. Die sofortige Verfügbarkeit dieser Informationen erleichtert die Koordination zwischen den verschiedenen Gewerken und kann Abstimmungsfehler reduzieren. In Kombination mit mobilen Endgeräten lassen sich administrative Tätigkeiten direkt vor Ort erledigen, sodass zeitaufwändige Nachbearbeitungen im Büro nicht mehr nötig sind. Aktuell zeigen sich klare Vorteile in den Bereichen der Rapportierung.

Der zweite wichtige Faktor, die Datenqualität der erfassten Eingaben wird derzeit als gut bis zufriedenstellend eingeschätzt. Dennoch ist klar, dass dieser Aspekt kontinuierlich überwacht und optimiert werden muss. Gemäss Aussagen in der Fokusgruppe ist eine hohe Datenqualität nämlich nicht nur für den unmittelbaren Projekterfolg, sondern auch für spätere Auswertungen (Controlling), Qualitätsnachweise und die strategische Planung wichtig.

Mit dem Einsatz dieser Technologie kann die Entwicklung von Anwendungen breiter auf die vorhandenen Personalressourcen verteilt werden, da sie nicht ausschliesslich auf spezialisierte Softwareentwickler angewiesen ist. Durch die Einbindung von Fachkräften ohne vertiefte Programmierkenntnisse wird praxisnahes Wissen direkt in die Entwicklung integriert, was in der Praxis zu passgenaueren Lösungen führt. Gleichzeitig wird die IT-Abteilung entlastet, sodass sie sich auf komplexe Integrations- und Architekturaufgaben konzentrieren kann.

Auf Anpassungen und Erweiterungen kann das direkt involvierte Fachpersonal dank der Flexibilität von Low-Code rasch reagieren. Dadurch wird die Reaktionsfähigkeit des Unternehmens gegenüber neuen Anforderungen im Projekt deutlich erhöht. Durch die kontinuierlichen Anpassungen wird auch das Interesse der Mitarbeitenden an Prozessoptimierungen geweckt, was immer wieder zu spannenden Ideen für weitere Low-Code-Applikationen führt.

Auf Unternehmensführungsebene spielt der Return on Investment (ROI) im Normalfall eine entscheidende Rolle, in diesem Fall eine eher untergeordnete Rolle. Stattdessen steht der unmittelbare operative Nutzen im Vordergrund. Die Zeitersparnis bei der Entwicklung und Anpassung von Anwendungen ist dabei einer der grössten Vorteile. Gerade in Bauprojekten, in denen Entscheidungen oft kurzfristig getroffen werden müssen, wirkt sich eine schnelle Bereitstellung digitaler Lösungen unmittelbar auf die Effizienz der Abläufe aus. Der Nutzen wird daher eher indirekt ersichtlich. Weiter zeigt die Erfahrung, dass einmal entwickelte Applikationen mit kleinen projektspezifischen Anpassungen häufig auch für andere Projekte wiederverwendet werden können. Der Ansatz dabei ist aber nicht von Beginn weg eine umfassende Applikation zu entwickeln, sondern die Vorteile einer schrittweisen und iterativen Entwicklung zu nutzen.

Eher schlecht wird aktuell die Integrationsfähigkeit von Low-Code in unternehmensübergreifende Prozesse bewertet. Meist werden nur wenige Schnittstellen zu anderen Systemen umgesetzt, wodurch die nahtlose Einbindung von Daten aus unterschiedlichen Quellen erschwert wird. Diese Fähigkeit ist jedoch wichtig, um Medienbrüche zu vermeiden und den Informationsfluss zu verbessern. Obwohl dies nicht zur Hauptstärke von Low-Code zählt, bleibt dieser Aspekt strategisch relevant.

Der zentrale Anwendungsfall dieser Forschung ist die Integration von IFC-Datenmodellen in eine Low-Code Applikation. Die Expertinnen sehen den Einsatz von Low-Code in Verbindung mit IFC derzeit als weniger sinnvoll, was vor allem auf technische Einschränkungen und das komplexe Format zurückzuführen ist. Konkreter wird die Anwendungsfall spezifische Datenextraktion und -integration hervorgehoben. Die Einbindung von IFC-Daten auf Baustellen ist jedoch grundsätzlich wichtig und könnte viele Unternehmenseigenen Standardisierungen verhindern und die Zusammenarbeit mit anderen Gewerken könnte auch vereinfacht werden.

5.2 Konkrete Anwendungsszenarien

Mithilfe qualitativer Interviews wurde untersucht, wie Unternehmen Low-Code-Technologien einsetzen, um ihre Prozesse effizienter zu gestalten. (ausführliche Interviews siehe Anhang G bis Anhang O) Nachfolgend wird eine Auswahl an konkreten Anwendungsszenarien dargestellt. Die zeigen, wie sich mit Low-Code durchgängige neue Prozesse umsetzen lassen. Im Fokus standen die Entwicklung von Applikationen, der Aufbau von Dashboards und die Automatisierung von Arbeitsabläufen. Zudem wurde betrachtet, wie die technische Umsetzung dieser Lösungen in den jeweiligen Unternehmenskontext integriert werden konnten.

5.2.1 Applikationen

Auf der Baustelle ergeben sich für ausführende Betriebe des Bauhaupt- und Baunebengewerbes sehr spannende Anwendungsfälle. So kann etwa eine Applikation zur Materialbestellung entwickelt werden, mit der benötigte Mengen direkt vor Ort erfasst und in Echtzeit an das Lager oder an Lieferanten übermittelt werden. Oder es können etwa Anwendungen für digitale Checklisten bei Sicherheitsbegehungen oder zur Meldung von Gefahrenstellen über das Smartphone genutzt werden.

Dokumentation während dem Sicherheitsrundgang

Arbeitsunfälle und die Sicherheitsprävention sind auf Baustellen nach wie vor ein sehr wichtiges Thema und erfordern immer ihre Aufmerksamkeit. Aus diesem Grund führen die Sicherheitsbeauftragten in der Regel alle zwei Monate umfassende Sicherheitsrundgänge durch. Ziel dieser Rundgänge ist es, potenzielle Gefahrenquellen frühzeitig zu identifizieren, die Einhaltung der Sicherheitsstandards zu überprüfen und bei Bedarf geeignete Massnahmen einzuleiten, um Unfälle zu vermeiden.

Ursprünglich erfolgte die Dokumentation dabei in einer Excel-Liste. Diese wurde allerdings nicht direkt während des Rundgangs, sondern erst im Anschluss am Schreibtisch ausgefüllt. Das hatte zur Folge, dass Beobachtungen und Notizen zunächst handschriftlich oder aus dem Gedächtnis festgehalten und anschliessend nachträglich in die Datei übertragen wurden. Dieses Verfahren war nicht nur zeitaufwändig, sondern auch fehleranfällig, da Informationen leicht vergessen/verloren gehen oder unvollständig dokumentiert werden konnten. Um diesen Prozess zu verbessern, wurde eine App entwickelt, die den Sicherheitsbeauftragten unmittelbar während des Rundgangs unterstützt. Anstatt die Ergebnisse später nachzutragen, kann der Verantwortliche direkt auf der Baustelle sämtliche Punkte digital erfassen. Die App führt ihn dabei durch eine strukturierte Checkliste, in der er Fragen beantwortet und einzelne Aspekte systematisch abhakt. Dadurch wird sichergestellt, dass keine Prüfpunkte vergessen werden, da die Erfassung standardisiert erfolgt.

Ein zusätzlicher Vorteil ist die darauf folgende Weiterbearbeitung im Büro. Die erfassten Daten sind webbasiert verfügbar und können von den Sicherheitsbeauftragten selbst oder von anderen Mitarbeitenden eingesehen, ergänzt und bei Bedarf als Bericht weitergeleitet werden. Dies vereinfacht die Nachbereitung markant und ermöglicht eine schnelle Kommunikation mit allen relevanten Stellen. So können Sicherheitsbeauftragte unmittelbar erkennen, in welchen Bereichen Handlungsbedarf besteht und wo bereits ein sicheres Umfeld gewährleistet ist, und dies durch Analysen und abgeleiteten Monitoring-Massnahmen ergänzen.

Weiter beinhaltet die Applikation eine Erfassung der auf der Baustelle anwesenden Personen. Ihre Namen können direkt aus der integrierten Adresskartei abgerufen und automatisch hinzugefügt werden. Über sie kann zeitgleich die Zugänglichkeit zur Baustelle geregelt werden. Der Polier hat somit jederzeit einen vollständigen Überblick über alle Beschäftigten, einschliesslich temporärer Arbeitskräfte, und verfügt dadurch über die relevanten Daten für die Einsatzplanung. (Schweizer, 2025)

Wertschöpfende/nicht Wertschöpfende Tätigkeiten

Diese Begriffe gehen auf die Grundsätze von Lean zurück. Darin gehören die sieben Arten der nicht Wertschöpfenden Tätigkeiten zur Verschwendung: Transport, Bestände, Bewegung, Wartezeit, Überproduktion, Überverarbeitung oder fehlerhafte Produkte, Ausschuss und Nacharbeiten. Im japanischen spricht man dabei von Muda (Fiedler, 2018, pp. 52–56).

In der Praxis brauchen die Arbeiter auf der Baustelle täglich viel Zeit für nicht Wertschöpfenden Tätigkeiten wie etwa das Transportieren von Materialien von einem Lagerplatz zum anderen, weil der ursprüngliche Lagerplatz für andere Arbeiten benötigt wird. Durch eine gezielte Planung kann diese Arbeit vermieden werden, indem der Lagerort von Anfang an klar festgelegt und allen Beteiligten bekannt gemacht wird. Selten fallen diese überflüssigen Arbeiten im Arbeitsalltag auf, da sie meist ungeplant und als Reaktion auf kurzfristige Entscheide geschehen. Dazu unterscheidet/klassifiziert der Polier die Anfallenden Tätigkeiten auf der Applikation in Wertschöpfende und nicht Wertschöpfende Tätigkeiten. Werden vermehrt nicht wertschöpfende Tätigkeiten festgestellt, werden diese genauer analysiert, gebündelt ausgeführt und nach Möglichkeit vermieden. (Pfister, 2024)

Materialbestellungen zur Fahrbahnerneuerung

Bei der Bestellung von Bauteilen für Fahrbahnerneuerungen wurde über einen längeren Zeitraum klassische Excel-Liste verwendet. Diese Datei diente als zentrales Hilfsmittel, um sämtliche benötigten Elemente zu erfassen und im weiteren Verlauf zu bestellen. Die Vorgehensweise war dabei stets gleich: Für jedes neue Projekt wurde die bestehende Excel-Datei aus einem vorangegangenen Auftrag kopiert und entsprechend angepasst. Dadurch

konnte zwar auf bereits bestehende Strukturen zurückgegriffen werden, gleichzeitig führte dieses Vorgehen jedoch auch zu einer Vielzahl von Dateiversionen, die projektweise getrennt abgelegt waren. Transparenz, Nachvollziehbarkeit und eine einheitliche Datenhaltung waren somit nur eingeschränkt gewährleistet.

Um diesen Prozess zu optimieren, wurde die Excel-Liste in eine Applikation überführt. Diese ermöglicht es, die Bestellung der benötigten Elemente direkt in digitaler Form zu erfassen. Die Daten werden dabei nicht mehr lokal in einzelnen Dateien gespeichert, sondern zentral in einer SharePoint-Datenbank hinterlegt. Auf diese Weise können die erfassten Informationen unmittelbar verarbeitet, systematisch ausgewertet und jederzeit digital eingesehen werden. Gleichzeitig besteht die Möglichkeit, die zusammengestellte Bestellung automatisch per E-Mail an die entsprechenden Fachstellen zu senden.

Ein weiterer wesentlicher Vorteil ist, dass sämtliche Daten aller Projekte nun zentral gespeichert und für alle berechtigten Nutzerinnen und Nutzer zugänglich gemacht werden. Anstelle einzelner, isolierter Excel-Dateien, die nur in einem bestimmten Kontext verfügbar waren, können die Informationen nun projektübergreifend und von allen Beteiligten eingesehen werden. (Bosshart, 2025)

5.2.2 Dashboards und Automatisierungen

Wie in vielen Bereichen sind Dashboards auch im Bauwesen ein wirkungsvolles Instrument, um Projektinformationen übersichtlich aufzubereiten und für alle Beteiligten leicht verständlich darzustellen. Sie ermöglichen die Zusammenführung von Daten aus verschiedenen Quellen in Echtzeit und machen so komplexe Sachverhalte auf einen Blick sichtbar.

Ein Dashboard kann beispielsweise im Rahmen einer Variantenstudie eingesetzt werden, um unterschiedliche Planungsoptionen miteinander zu vergleichen und ihre Auswirkungen auf Kosten, Termine und Ressourcen anschaulich gegenüberzustellen. Ebenso lassen sich Aspekte der Modellqualität transparent darstellen. Beispielsweise kann die Einhaltung von Namenskonventionen, der Detaillierungsgrad von Bauteilen oder die Vollständigkeit der Informationen im BIM-Modell überprüft und visualisiert werden.

Aushubkubatur berechnen

Der Aushub ist einer der kostenintensivsten und logistisch anspruchsvollsten Arbeitsschritte in Bauprojekten. Neben den reinen Entsorgungskosten verursacht er ein erhebliches Verkehrsaufkommen, da grosse Mengen Material mit Lastwagen von der Baustelle abtransportiert werden müssen. Um die damit verbundenen Aufwände im Hinblick auf die Kosten und die logistischen Ressourcen möglichst präzise planen zu können, ist eine exakte Ermittlung der Aushubmenge unerlässlich. Traditionell erfolgte die Abschätzung des Aushubs häufig auf Basis von Erfahrungswerten oder vereinfachten Berechnungen. Dies führte jedoch nicht selten zu Unsicherheiten, die in der späteren Bauausführung zu Mehrkosten oder Engpässen in der Transportlogistik führen konnten.

Mithilfe des Low-Code/No-Code-Programms *Rhino-Grasshopper* lässt sich die Aushubkubatur auf Basis eines 2D-Planlayouts in Kombination mit einem ergänzenden 3D-Geländemodell bestimmen. Das Programm ermöglicht es, die vorhandenen Planungsdaten in eine räumliche Geometrie zu überführen und daraus die exakten Volumina des anfallenden Aushubs zu berechnen. Dieser digitale Ansatz ist nicht nur wesentlich genauer, sondern erlaubt auch eine schnelle Anpassung, falls sich im Verlauf des Projekts Änderungen am Geländeprofil oder an der Baugrube ergeben.

Die gewonnenen Daten bilden die Grundlage für eine Vielzahl von Planungsentscheidungen. Einerseits können damit die zu erwartenden Kosten für den Aushub kalkuliert werden. Andererseits lässt sich auf Basis der Volumina die benötigte Anzahl an Lastwagenfahrten ableiten, was eine gezielte Planung des Verkehrsaufkommens und der Transportlogistik ermöglicht. Zusätzlich liefert die Berechnung wertvolle Informationen über die Dimensionierung von Lagerflächen, etwa für die temporäre Zwischenlagerung der Humusschicht, die im Zuge der Erdarbeiten gesondert behandelt wird. (Schweizer, 2025)

Fortschritt und Qualität des Modells messen und sichtbar machen

Die frühen Planungsphasen eines Bauprojekts sind sehr zeitintensiv. Sie erfordern zahlreiche Abstimmungen und eine stetige Verfeinerung der Projekthinhalte. Für Aussenstehende bleibt dieser Prozess jedoch meist unsichtbar, da viele Entwicklungen im Hintergrund stattfinden und nicht unmittelbar greifbar sind. Besonders die detaillierte Ausarbeitung eines digitalen Bauwerksmodells ist sehr aufwendig und wirkt sich auf die Qualität, Genauigkeit und Strukturierung der Modellinformationen aus.

Ein Dashboard kann hier entscheidend zur Transparenz beitragen. Mit einer anschaulichen Visualisierung wird der Fortschritt der Modellierung, die Qualität der erstellten Bauteile und die Einhaltung definierter

Namenskonventionen nachvollziehbar dargestellt. Damit kann Klarheit für alle Projektbeteiligten geschaffen werden, was zugleich die Steuerung und Qualitätssicherung im digitalen Bauprozess unterstützt (Vesterlund & Falk, 2025).

Variantenstudien mit Kosten

In frühen Planungsphasen eines Bauprojekts werden die Grundlagen für das spätere Bauwerk gelegt. Erste Ideen werden schrittweise konkretisiert, Entwürfe verfeinert und zahlreiche Varianten miteinander verglichen. Dies erfordern einen hohen zeitlichen Aufwand sowie eine enge Abstimmung zwischen Architekten, Ingenieuren, Fachplanern und Auftraggebern. Aus kleinen Anpassungen können über den gesamten Bauprozess grosse Auswirkungen nach sich ziehen. Auch hier bleibt dieser Prozess jedoch oft unsichtbar, da die Vielzahl an Diskussionen, Abstimmungen und Anpassungen im Hintergrund stattfindet und nur selten in einer klar erkennbaren Form dokumentiert sind.

Die sukzessive Erstellung und Detaillierung des digitalen Bauwerksmodells (BIM-Modell) ist besonders aufwendig. Jeder Bauteil muss präzise erfasst, mit den richtigen Parametern versehen und entsprechend der definierten Standards strukturiert werden. Dieser iterative Prozess beeinflusst die Qualität, Genauigkeit und Konsistenz der erhaltenen Informationen massgeblich. Gleichzeitig steigt mit zunehmender Komplexität des Modells auch die Gefahr, dass Fehler oder Unstimmigkeiten unbemerkt bleiben und erst in späteren Projektphasen sichtbar werden. Ein Dashboard schafft hier mehr Transparenz und Übersichtlichkeit in dem der Fortschritt der Modellierung in anschaulicher Form visualisiert wird. Neben quantitativen Kennzahlen zu: Kosten, Materialien, Mengenauszügen können auch qualitative Aspekte wie der Wärmedurchgangskoeffizient oder der prognostizierte Energieverbrauch aufgezeigt werden.

Für alle Projektbeteiligten vom Planer über den Bauherrn bis hin zu externen Stakeholdern schafft das Dashboard somit eine gemeinsame Sichtweise auf das digitale Bauwerksmodell. Die Nachvollziehbarkeit der getroffenen Entscheidungen wird erhöht und fördert im selben Zuge die faktenbasierte Problembewertung und -lösungsfindung. Weiter unterstützt es das Projektteam aktiv bei der Qualitätssicherung. (Bosshart, 2025)

Archivierung von Projektunterlagen und Datenordnern

Gemäss der SIA-Norm sind Planer und Unternehmer dazu verpflichtet, bis zu zehn Jahre nach Abschluss eines Bauprojekts auftretende versteckte Mängel zu beheben. Eine sorgfältige und übersichtliche Archivierung sämtlicher Projektunterlagen ist daher die Basis, um diese Verpflichtung umsetzen zu können. In der Praxis zeigt sich jedoch häufig ein anderes Bild: Sobald ein Projekt abgeschlossen ist, werden die personellen und organisatorischen Ressourcen unmittelbar in neue Bauvorhaben investiert. Die Archivierung der Dokumente erfolgt oft manuell, wird aufgeschoben oder sogar vergessen. Dies führt zwangsläufig zu Intransparenz und kann im Ernstfall, etwa bei der Durchsetzung von Gewährleistungsansprüchen, erhebliche Risiken für alle Beteiligten und vor allem für das Unternehmen mit sich bringen.

In diesem Elektrounternehmen schafft die eigens aufgebaute Automatisierung mittels Low-Code-Applikation Abhilfe. Mit nur einem Knopfdruck werden sämtliche relevanten Unterlagen automatisch aus dem operativen Betriebssystem extrahiert, ordnerweise aufbereitet und strukturiert am richtigen Ort innerhalb der Serverarchitektur zur Archivierung abgelegt. Damit wird sichergestellt, dass diese Unterlagen nicht mehr manipuliert werden können und diese auch in zehn Jahren ohne Branchensoftware geöffnet werden können. Somit sind alle Projektdaten vollständig, geordnet und jederzeit abrufbar. Neben der Zeitersparnis erhöht sich dadurch auch die Rechtssicherheit, da die Nachweispflicht im Falle von Mängeln problemlos erfüllt werden kann. Gleichzeitig entlastet die Automatisierung die Personalressourcen der Projektteams und verhindert, dass wichtige Informationen im hektischen Übergang von einem Projekt zum nächsten verloren gehen. (Blessing & Asairatnam, 2025)

Automatische Deckblätter

Deckblätter folgen in grossen Unternehmen festen Richtlinien und sollen eine Wiedererkennung haben. Sie dienen nicht nur der formalen Einheitlichkeit, sondern auch der unternehmerischen Gestaltungsrichtlinien. In der Praxis werden die Word-Vorlagen für diese Dokumente jedoch häufig aus bereits bestehenden Projekten kopiert und anschliessend manuell angepasst. Mit der Zeit führt dieses Vorgehen zu immer grösseren Abweichungen vom ursprünglichen Standard. Das manuelle Ausfüllen der projektspezifischen Informationen wie Projektname, Verantwortliche, Datum oder Version ist zudem fehleranfällig und kann leicht zu Inkonsistenzen oder Ungenauigkeiten führen.

Anstatt die Vorlagen manuell anzupassen, können die relevanten Informationen direkt aus der Branchensoftware also der zentralen Projektdatenbank abgefragt und automatisch in das Deckblatt integriert werden. Dadurch ist

sichergestellt, dass alle Dokumente stets nach den geltenden Richtlinien erstellt werden und sie eine klare Wiedererkennbarkeit aufweisen. Ihre Inhalte sind immer korrekt und entsprechen den aktuellen Datenstand. Gleichzeitig reduziert sich der manuelle Aufwand erheblich da nur noch ein System als «single point of truth» (SPOT) für den Daten und Informationsunterhalt gepflegt werden muss. (Blessing & Asairatnam, 2025)

5.2.3 Technische Umsetzung

Damit die Low-Code Technologie in der Branche eingesetzt werden kann, sind die technische Voraussetzungen massgebend. Innovatoren der Baubranche empfehlen, auf bereits gemachte Erfahrungen anderer Unternehmen zurückzugreifen um von ihren Herausforderungen und den daraus resultierenden Lernkurven profitieren zu können. (Noack & Fedtke, 2025).

Meist verwendet Plattformen in der Praxis

Anlässlich dieser Studie wurde eine qualitative Stichprobenumfrage mit neun Teilnehmenden (n=8) durchgeführt. Der Stichprobenumfang ist sehr gering und daher mit einer gewissen Vorsicht zu geniessen. Für das Ableiten einer ersten Tendenz können diese Ergebnisse jedoch bereits erste Erkenntnisse bieten.

Aus den geführten Interviews geht hervor, dass die Microsoft Power Plattform mit fünf Nennungen die am häufigsten verwendete Low-Code-Plattform ist (Blessing & Asairatnam, 2025; Bosshart, 2025; Pfister, 2024; Schweizer, 2025; Vesterlund & Falk, 2025). Dynamo wurde zweimal (Blessing & Asairatnam, 2025; Vesterlund & Falk, 2025) und UI-Path sowie Grasshopper je einmal (Schweizer, 2025) erwähnt. Siehe Abbildung 37.

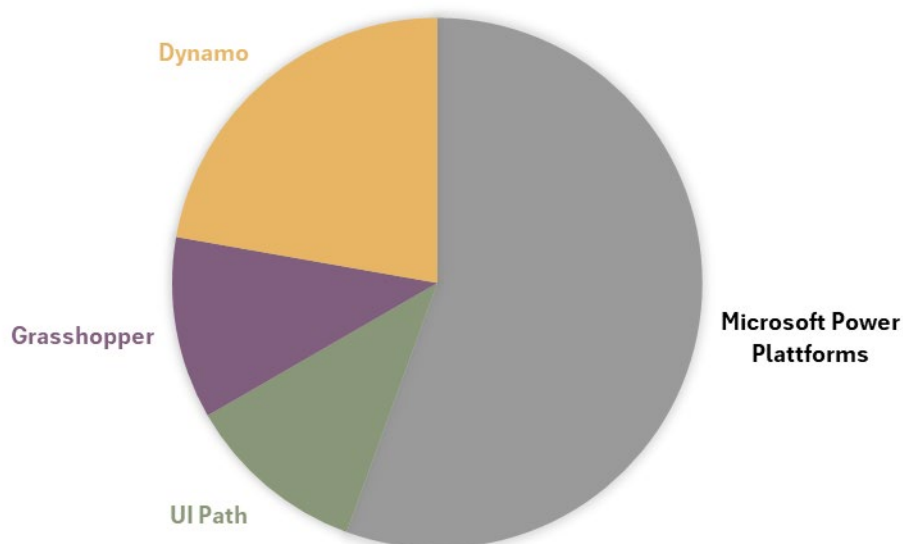


Abbildung 37; Verwendete Low-Code-Plattformen in der Baubranche auf der Basis von Expertenaussagen (eigene Abbildung)

Datenqualität und Datenintegration

Zu den Themen: Dateninput, Datenoutput, Datenqualität, Data Governance und der Wichtigkeit der Datendurchgängigkeit verfolgen die Unternehmen unterschiedliche Ansätze. Teilweise setzen sie auf Schulungen der Mitarbeitenden (Blessing & Asairatnam, 2025; Pfister, 2024) teilweise auf klar definierte Zugriffsrechte (Oberholzer, 2025) oder auf die Verantwortlichkeit spezialisierter IT-Teams (Blessing & Asairatnam, 2025; Bosshart, 2025). Eignigkeit besteht jedoch darin, dass es eines der wichtigsten Themen ist und zu grossen Herausforderung führen kann.

So entwickelte ein Softwareunternehmen aus der Schwierigkeit der Datenintegration aus unterschiedlichsten Softwareprodukten und dem Life Cycle Data Management (LCDM) den «Swiss LCDM-Hub». Dieses System dient der Vereinheitlichung und zentralen Verwaltung von Daten über den gesamten Lebenszyklus hinweg und ermöglicht dadurch eine effizientere Nutzung und Pflege der Informationen. Als Beispiel wird erwähnt, dass in grossen Spitalbauten ist die Datendurchgängigkeit ein grosses Thema für die optimierte Bewirtschaftung ist und die Wirtschaftlichkeit des Betriebs wesentlich beeinflusst.

Während der Bauzeit entstehen kontinuierlich grosse Mengen an Daten, die für die Planung, Ausführung und spätere Nutzung eines Gebäudes zentral sind. Diese reichen von technischen Spezifikationen und

Materialangaben bis hin zu Protokollen, Abnahmeberichten und BIM-Modellen. In der Praxis gehen jedoch viele dieser wertvollen Daten verloren, da sie lokal auf einzelnen Rechnern gespeichert werden oder nur für bestimmte Planungsbeteiligte zugänglich sind. Dadurch entstehen Lücken in der Dokumentation, die einen effizienten Betrieb und eine nachhaltige Bewirtschaftung des Gebäudes erschweren können. Der Swiss LCDM-Hub stellt sicher, dass alle während der Planungs- und Bauphase gesammelten Daten in der richtigen Granularität und Genauigkeit strukturiert aufbereitet und den relevanten Beteiligten über den gesamten Lebenszyklus des Gebäudes hinweg zur Verfügung stehen.

Im Unterschied zu den zuvor beschriebenen Low-Code-Applikationen ist der Swiss LCDM-Hub eine vom Software-Unternehmen hardcodierte Plattform. Gleichzeitig bietet sie den Plattformnutzenden die Möglichkeit, ihre Daten flexibel nach dem Low-Code-Ansatz selbst zusammenzustellen und an projektspezifische Bedürfnisse anzupassen. Damit verbindet die Plattform die Stabilität einer robusten Kernarchitektur mit der Agilität, individuelle Lösungen ohne tiefgreifendes Programmierwissen aufzubauen. (Oberholzer, 2025)

5.3 Chancen und Herausforderungen aus Bau-Expertensicht in der Schweiz

Mithilfe der qualitativen Interviews mit Experten aus verschiedenen Bereichen der Schweizer Baubranche, mit Schwerpunkt auf der Ausführung von Baumeisterarbeiten, wurden die vielfältigen Chancen und Herausforderungen von Low-Code erfasst. Ein zusätzlicher Blick über die Landesgrenzen sowie ein Expertengespräch mit einem Planungsbüro aus Dänemark ergänzen diese Datenerhebung. Die vollständigen Interviews sind in Anhang G bis Anhang O dokumentiert. Die befragten Unternehmen wurden anhand eines Venn-Diagramms (teilanonymisiert) den drei Sphären Bautechnik, Planung und Management sowie Technik zugeordnet (siehe Abbildung 38).

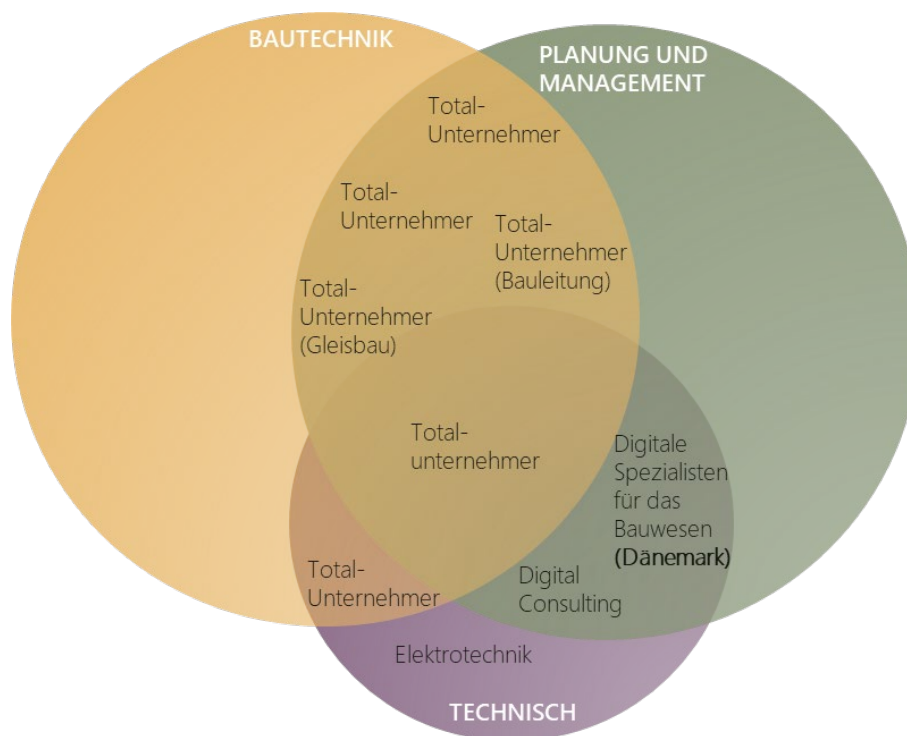


Abbildung 38; Venn-Diagramm der Arbeitsgattungen (eigene Darstellung)

Bei allen Expertengesprächen wurde über die zwei nachfolgenden Fragen diskutiert:

- Welche **Chancen** bietet der Einsatz von Low-Code Technologie im Vergleich zu **herkömmlichen Programmiermethoden** im Bauwesen?
- Was sind die **Herausforderungen** und möglichen Risiken bei der Integration von Low-Code-Technologien in den Arbeitsalltag im Bauwesen

Dabei wurden nachfolgende Erkenntnisse zu den Chancen und Herausforderungen zur betrieblichen Anwendung von Low-Code gewonnen.

Einfluss auf die Skalierung

In mehreren Grossorganisationen bleiben Low-Code-Automatisierungen als Inselösungen in den Fachbereichen stecken. Die entwickelten Anwendungen werden nur dort genutzt, wo sie entstanden sind, und nicht konzernweit verbreitet. Dies liegt oft daran, dass ein systematischer Know-how-Transfer fehlt. Wenn Prozesse und Best Practices nicht klar dokumentiert und weitergegeben werden, können Mitarbeitende oft nur schwer von einzelnen Pilotprojekten überzeugt werden (Bosshart, 2025). Um dieses Silodenken aufzubrechen, bedarf es einer konsequenten Standardisierung, die sowohl die Technologie als auch den gesamten Entwicklungs- und Betriebsprozess beinhaltet.

Eine erfolgreiche Digitalisierung mittels Low-Code setzt voraus, dass sämtliche Prozessschritte vollständig standardisiert werden. Von der Anforderungsanalyse und Modellierung bis hin zur Bereitstellung und Wartung (Bosshart, 2025; Pfister, 2024). Nur so lassen sich Automatisierungsbausteine effizient wiederverwenden und schneller in unterschiedlichen Fachabteilungen ausrollen. Einfache, allgemeine Automatisierungen können oft noch problemlos unternehmensübergreifend eingesetzt werden. Branchenspezifische Low-Code-Anwendungen stossen dagegen regelmässig an Grenzen, da sie mitunter aufwändige Anpassungen an unternehmensinterne Standards, Schnittstellen und Compliance-Richtlinien erfordern (Pfister, 2024). Daher beschränken viele Organisationen den Einsatz von Low-Code auf Bereiche, in denen sich unmittelbar hohe Umsätze und schnelle Skaleneffekte realisieren lassen, beispielsweise in wichtigen Prozessen mit hohem Transaktionsvolumen.

Einfluss von Nutzenden

Doch Technologie allein genügt nicht, um eine breite Akzeptanz zu erreichen. Zahlreiche Unternehmen stehen neuen Low-Code-Initiativen auch aus Angst vor dem Unbekannten skeptisch gegenüber. Einige Verantwortliche geben offen zu: «Wir verwenden kein Low-Code» (Terzic, 2025) oder «Das ist nichts für unser Unternehmen» (Terzic, 2025). Meist sind es nur die mutige Early Adopter, welche den Sprung wagen, während sich der Grossteil der Belegschaft lieber auf bewährte Verfahren verlässt (Bosshart, 2025). Dieser Widerstand entsteht häufig aus mangelndem Vertrauen in die Plattformen sowie aus dem Unbehagen, die Kontrolle über kritische Geschäftsprozesse an eine scheinbar einfache Oberfläche abzugeben und somit ersetzbar zu werden.

Um Skeptiker zu überzeugen, können erfahrene Anwender als Multiplikatoren gewonnen werden. Anhand praxisnaher Anwendungsfälle können sie dann demonstrieren, wie Low-Code den Arbeitsalltag erleichtert und repetitive Arbeit überflüssig macht. So nutzen im Bauwesen viele Planende bereits Common Data Environments (CDE), um Notizen oder Fotos zu teilen, statt E-Mails zu schreiben. (Vesterlund & Falk, 2025). Solche alltäglichen Beispiele zeigen, dass eine nahtlose Integration in bestehende Tools und Formate den Einstieg erleichtert und Hemmschwellen abbaut.

Parallel zur Überzeugungsarbeit sollte in die Aus- und Weiterbildung investiert werden (Blessing & Asairatnam, 2025). Zwar sind Low-Code-Plattformen bewusst so gestaltet, dass auch weniger stark geschulte Mitarbeitende erste Automatisierungen umsetzen können, doch für komplexere Szenarien und unternehmensweite Rollouts sind trotzdem weiterführende Schulungen und ein breites Wissen unverzichtbar. Schulungen benötigen zwar Zeit und Ressourcen, doch ihre Erfolge zahlen sich langfristig aus, indem sie die Fehlerquote senken und die Mitarbeiterzufriedenheit steigern. Blessing und Asairatnam empfehlen Unternehmen daher strukturierte Trainingsprogramme entwickeln und anwenden (Blessing & Asairatnam, 2025).

Zugriffsrechte und Datendurchgängigkeit

Mehrere Experten erklären, dass eine wirksame Data-Governance eng mit Schulung und Standardisierung verbunden ist (Blessing & Asairatnam, 2025; Bosshart, 2025). Häufige oder einmalige Datentransfers können zu Qualitätsverlust und Inkonsistenzen führen, wenn Rollen und Zugriffsrechte nicht klar definiert sind (Bosshart, 2025; Pfister, 2024). Im digitalen Bau etwa wächst die Komplexität, sobald Planer, Bauleiter oder Facility Manager unterschiedliche Lese und Schreibrechte benötigen. Zugriffsrechte können sehr schnell komplex werden, müssen aber trotz heterogener Gewerke überschaubar bleiben (Schweizer, 2025). Ein sorgfältiges Rollen und Rechteanagement sowie klare Prozesse für Rechteanträge und -änderungen sind die Grundlage dafür, dass Daten konsistent genutzt, aktualisiert und archiviert werden können (Blessing & Asairatnam, 2025).

Ein weiterer Kernpunkt ist die Datendurchgängigkeit. Die Schnittstellenanforderungen ändern sich fortlaufend und werden oft erst im Betrieb konkret. Deshalb ist es unerlässlich, Fachabteilungen wie das Datenmanagement zu befähigen, Interfaces selbstständig zu konfigurieren und anzupassen (Oberholzer, 2025). Nur so lassen sich nahtlose Datenflüsse zwischen verschiedenen Systemen sicherstellen und lästige Medienbrüche vermeiden. Einmal erstellte digitale Modelle sind per se skalierbar. Das heisst es kostet nicht mehr, wenn sie von vielen Leuten und Unternehmen genutzt werden, als wenn es nur ein paar wenige sind. (Vesterlund & Falk, 2025). Fehlende Schnittstellen lassen sich mit Low-Code rasch schliessen, sodass Altsysteme um modernste Funktionen ergänzt werden können und Updates oder Systemunterbrechungen weniger gravierende Auswirkungen haben (Blessing & Asairatnam, 2025; Vesterlund & Falk, 2025).

Das Lifecycle-Data-Management sollte demnach so gestaltet sein, dass Daten nicht nur einmal migriert, sondern kontinuierlich synchronisiert und gepflegt werden. Nur wenn das Qualitätsmanagement und agile Prozesse Hand in Hand gehen, kann die Nutzungsdauer von Daten aus dem Bau über Jahre hinweg gesichert und Vertrauen in ihre Verlässlichkeit geschaffen werden (Oberholzer, 2025).

Trotz dieser Potenziale gibt es weiterhin viele Herausforderungen bei der Digitalisierung. So führt die Fragmentierung der Plattformen durch die unterschiedlichen Paradigmen der unterschiedlichen Anbieter beispielsweise dazu, dass Fachbereiche mit mehreren Systemen und Softwareanbietern arbeiten müssen, um ihre Prozesse abzubilden. Je nach Anbieter können sechs oder mehr einzelne Tools parallel im Einsatz sein. Das erhöht den Gesamtaufwand und erschwert die Datendurchgängigkeit markant. Hier gilt es, durch unternehmensweite Integrationsstandards und projektübergreifende Vorgaben dem Wildwuchs zu begegnen und eine zentral koordinierte Steuerung zu etablieren (Vesterlund & Falk, 2025).

Reduzierung der Komplexität und Arbeitsweise

Ein wesentliches Versprechen von Low-Code ist die Reduzierung der Komplexität. Konkret werden komplexe Programmierlogiken auf eine für Business-Anwender verständliche Abstraktionsebene heruntergebrochen. Dadurch können auch Nutzer ohne IT-Hintergrund eigene Anwendungen entwerfen und anpassen (Schalko, 2025). Diese Demokratisierung der Softwareentwicklung kann die Innovationskraft im Unternehmen fördern und den Fachabteilungen eine neue Autonomie verleihen. Oftmals stellen Low-Code-Anwender in den Unternehmen eine Erweiterung zur IT-Abteilung dar (Blessing & Asairatnam, 2025; Bosshart, 2025). Zudem kann es für Unternehmen nachteilig sein, dass hardcodierte Codes bei Abwesenheit der Ersteller manchmal nicht mehr nachvollziehbar sind (Blessing & Asairatnam, 2025).

Im Prototyping erweist sich Low-Code als besonders leistungsstark, da sich mit Tools wie der Microsoft Power Platform Prozesse in Echtzeit modellieren, testen und modifizieren lassen. Dadurch ist es möglich, Funktionalitäten unmittelbar an veränderte Bedürfnisse anzupassen und Prototypen bereits in frühen Projektphasen zu validieren (Bosshart, 2025; Vesterlund & Falk, 2025). Die hohe Geschwindigkeit wirkt sich positiv auf die Motivation der Teams aus und macht Entwicklungsvorhaben spürbar flexibler und erfahrungsbasierter (Vesterlund & Falk, 2025). Allerdings ist Low-Code in grossen Unternehmen oftmals nur bedingt praktikabel, da die Plattformen und organisatorischen Strukturen häufig nicht auf eine flächendeckende Skalierung ausgelegt sind. Es gibt aber Firmen bei welchen Applikationen flächendeckend genutzt werden (Blessing & Asairatnam, 2025). Zwar finden kleinere Module und Automatisierungen schnell Verbreitung, doch wenn es darum geht, eine grössere Nutzeranzahl und einem weiteren Projekt zu koordinieren, stossen viele Nutzer mit den Datenschutzrichtlinien an ihre Grenzen (Oberholzer, 2025).

5.4 Alternative Low-Code-Plattform

Variante Autodesk Platform Service (APS)

Die Autodesk Platform Services (APS) ist eine Sammlung von Cloud-API. Mithilfe dieser cloudbasierten Dienste lassen sich Konstruktions- und Projektdaten aus Autodesk-Umgebungen wie: BIM 360, Autodesk Construction Cloud oder Fusion Team nutzen. Mithilfe dieser cloudbasierten Dienste werden Arbeitsabläufe in den Bereichen Architektur, Ingenieurwesen, Bauwesen und Fertigung digitalisiert. Design- und Konstruktionsdaten sind dann nicht mehr isoliert auf einzelnen Rechnern oder lokal verwalteten Speichern abgelegt, sondern zentral in der Cloud verfügbar.

Autodesk Platform Services dient dabei als verbindendes Element: Unterschiedliche Tools, Anwendungen und Beteiligte greifen über API auf dieselben Daten und Metadaten zu. Funktionen wie die Versionierung, die Änderungsverfolgung oder die Zugriffskontrolle werden systematisch unterstützt. Dadurch entstehen durchgängige

und nachvollziehbare Prozesse. Das von der Konzeptphase über das Design und die Konstruktionsplanung bis hin zur Fertigung oder Errichtung. Wesentlich sind dabei Interoperabilität, Automatisierung und die modulare Erweiterbarkeit der Systeme. Dadurch können Unternehmen ihre Planungs-, Prüf- und Produktionsprozesse effizienter, konsistenter und kollaborativer gestalten. Über die Data Management API stellt Autodesk den standardisierten Zugriff auf Dateien, Versionen und Metadaten in unterschiedlichen Cloud-Diensten bereit. (Autodesk Inc., 2025b, 2024)

Beispiel einer unternehmenseigenen APS-Plattform

Als Beispiel hat sich ein Planungsunternehmen auf APS diverse eigene Applikationen erstellt. Diese sind auf einen spezifischen Anwendungsfall ausgerichtet und ermöglichen dadurch eine effiziente Durchführung der jeweiligen Tätigkeiten. Das Planungsunternehmen hat beispielsweise Applikationen für einen: *Spezialplanung-Simulator*, *IFC Colour Converter*, *Bauablaufsimulator*, *Quality Checker*, *Bauvorschritts Simulator*, *Datenkonversationstool für spezial CAD*, *Vergleich Datenbank mit ACC* oder einen *Bauetappensimulator*. (siehe Abbildung 39)

Diese Applikationen wurden jeweils von Mitarbeitenden mit den von APS bereitgestellten Modulen und Funktionen erstellt. Nach mehreren Testläufen, der Freigabe durch Vorgesetzte und der Aktivsetzung werden sie als benutzerfreundliche Applikationen allen Mitarbeitenden zur Verfügung gestellt. Ab diesem Zeitpunkt können diese standardisierten Prozesse in verschiedenen Projekten genutzt werden.

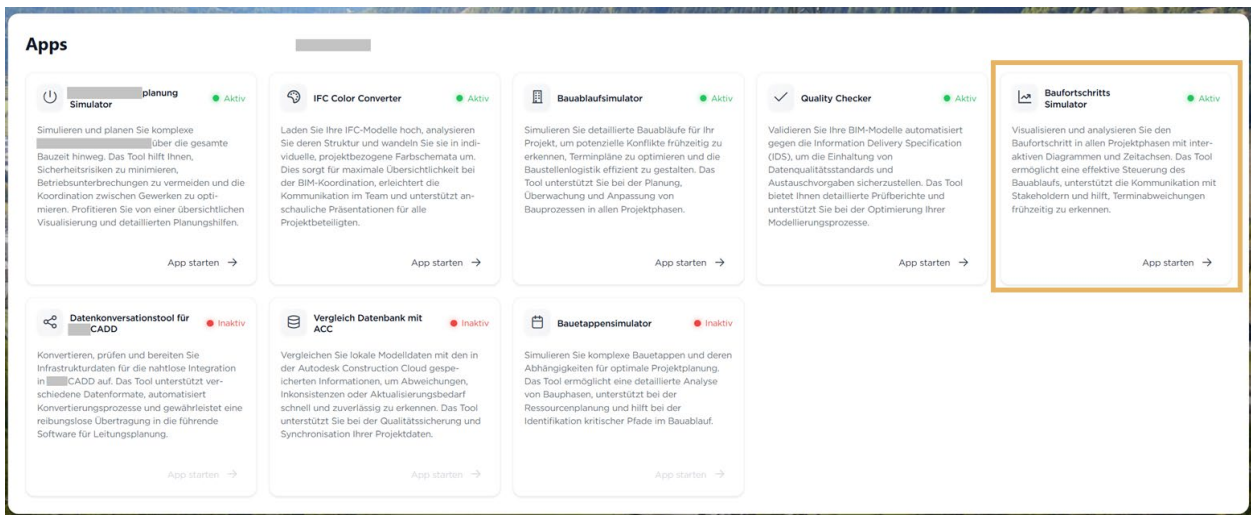


Abbildung 39; Auszug den unternehmenseigenen Apps aus APS (eigene Abbildung)

Applikation zur Bauvorschritts-Simulation

In der APS-Applikation «Bauvorschritts Simulator» können Bauleitende ihre Bauausführung schrittweise simulieren. (siehe Abbildung 40) Dazu wird zunächst die projektspezifische IFC-Datei sowie das mit Microsoft Project erstellte Terminprogramm geladen. Anschliessend aktualisieren sich automatisch die Anzeigen im Viewer 1 sowie das Bauprogramm im unteren Bereich der Oberfläche.

In einem weiteren Schritt können die Planenden ein zusätzliches, detailliertes Terminprogramm für einen speziellen Bauabschnitt hochladen. Per Knopfdruck aktualisiert sich danach die Ansicht im Viewer 2. Dabei wird ersichtlich, ob die eingeplanten Termine und Zeiten mit den vorgegebenen Planwerten übereinstimmen. Objekte, die grün markiert sind, entsprechen den Vorgaben. Rot markierte Objekte weisen Terminüberschneidungen auf und müssen entsprechend angepasst werden.

(Die integrierten Bauwerksmodelle in Viewer 1&2 sowie einige Angaben im Bauablauf wurden aus Datenschutzgründen absichtlich anonymisiert.)

Diese Applikation bietet den Bauleitenden eine sehr gute Unterstützung bei der terminlichen Koordination zwischen verschiedenen Gewerken. Durch die visuelle Darstellung der beiden terminverknüpften IFC-Dateien kann in Besprechungen in der Regel rasch eine geeignete Lösung gefunden werden.

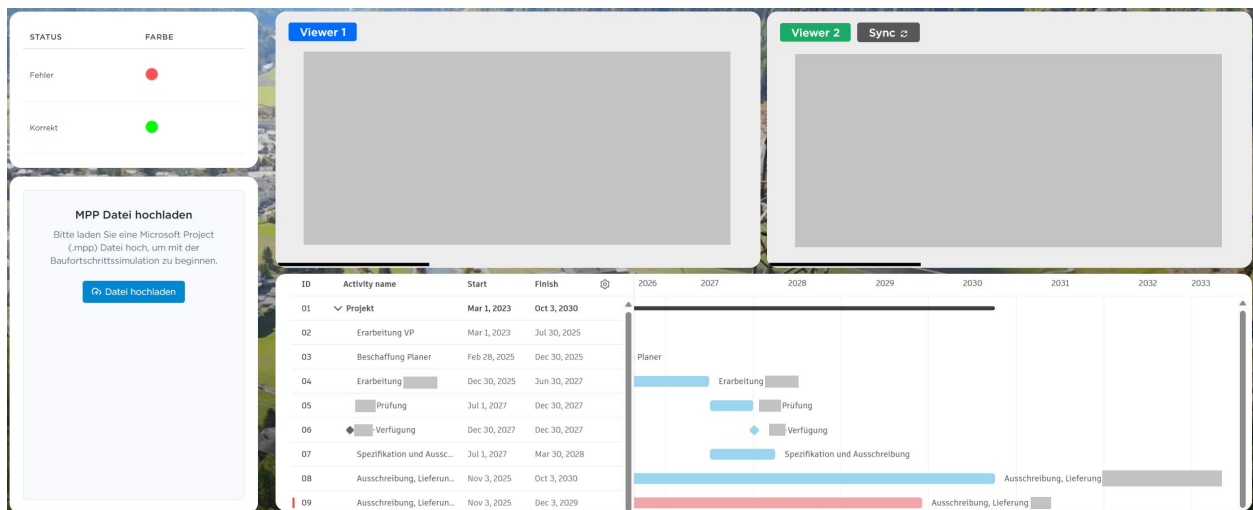


Abbildung 40; Auszug der unternehmenseigenen Applikation «Baufortschritts Simulator» aus APS (eigene Abbildung)

6 Diskussion

In der Diskussion werden die erzielten Ergebnisse kritisch reflektiert und in Bezug zum Stand der Forschung gesetzt. Ziel dabei ist es besser zu verstehen, welche Ansätze im Zusammenhang mit Low-Code und der Applikation in Verbindung mit IFC technisch möglich und im Arbeitsalltag sinnvoll sind.

6.1 Technische Voraussetzungen mit IFC

Wie vielerorts stellt auch die Einbindung und Weiterverarbeitung von IFC-Daten in der Praxis eine grosse Herausforderung dar. Die Daten eines IFC-Dokumentes lassen sich einerseits mittels einer Autoren-Software, andererseits mit einem Text-Editor einsehen und auch bearbeiten. Gerade die zweite Variante bedingt aber Grundkenntnisse in der Modellstruktur des IFC-Datenschemas (Schildknecht, 2023, pp. 2–38). Zusätzlich kann die verwendete Low-Code-Plattform von Microsoft IFC-Dateien nicht direkt lesen. Aus diesem Grund wurde beim Aufbau des Prototyps zunächst entschieden, den Datenzugriff auf die IFC-Datei mittels Python-Code zu verarbeiten und die Bauwerksdaten in einer CSV-Datei zwischenspeichern. Anschliessend wird diese als Excel-Datei gespeichert, wodurch ein neutrales Format entsteht. Die Verwendung der Excel-Datei ermöglicht allen Beteiligten einen einfachen Zugriff auf die Daten, ohne dass zusätzliche Software erforderlich ist. Mit der zentralen Ablage auf SharePoint wird diese danach zur gemeinsamen Datenbank aller Beteiligten.

Für die Übersichtlichkeit des Modelles ist eine Visualisierung sicher zwingend ([1], Fokusgruppe, 2025a). Dabei macht aber die allgemeine Menge an Bauteilen eine Bearbeitung am App sehr schwierig ([5], Fokusgruppe, 2025a). Auch ist die vom IFC-Datenschema vorgesehene Fertigstellung eines Bauteils in Prozent in der Realität zu prüfen ([7], Fokusgruppe, 2025a). Zwar gibt es Möglichkeiten IFC-Modelle in einer App einzubetten, diese bedingen aber eine zusätzliche Speicherung auf einer weiteren Dokumenten-Cloud. Als Beispiel werden hier Speckle oder der PCF-Component herangezogen.

Im selbst umgesetzten Prototyp auf der Basis von Microsoft, wird die transformierte IFC-Datei zuerst in einem Dashboard von PowerBI eingebettet, bevor diese Darstellung über die Zusatzapplikation von Tracer in der Low-Code-Applikation angezeigt wird. Es werden unnötige Datenflüsse und x-fache Transformationen geschaffen, welche zu Fehlern und zu Unübersichtlichkeit führen und der Ansatz der einzigen Datengrundlage wird auch verlassen. Auch bei Alternativen wie Speckle und PCF-Component sind zusätzliche Kompetenzen, etwa im traditionellen Codieren, für den Umgang mit externen Engines und API-Brücken erforderlich.

Die Einbindung eines IFC-Dokumentes ist in Low-Code mit einer Applikation als schwierig zu betrachten. Zwar ist das Lesen und Anzeigen möglich, das Bearbeiten, Schreiben und vor allem das Extrahieren von Metadaten setzt jedoch voraus, dass die geänderten Informationen in die Datenbank zurückgeschrieben und in die Struktur des IFC-Datenschemas übertragen werden.

Für den Prototyp sind mehrere Schnittstellen notwendig, die jedoch alle innerhalb desselben Software-Universums von Microsoft liegen, was die Integration deutlich vereinfacht. Dennoch stellen Software-Updates und

Änderungen in den Datenschutzrichtlinien eine erhebliche Hürde für die Anwendung dar, da im schlimmsten Fall kein Zugriff auf die Daten möglich ist. Zudem kann die Leistungsfähigkeit der Infrastruktur eines Unternehmens, je nach Dateigrösse vom digitalen Bauwerksmodells, an ihre Grenzen stossen und den Verarbeitungsprozess negativ beeinflussen ([5], Fokusgruppe, 2025a).

6.2 Integration einer IFC-Datei im Alltag

Das IFC-Datenschema ermöglicht den Austausch eines standardisierten Datenformats und trägt so zur Interoperabilität verschiedenster Planungssoftwares bei. Der Begriff, der unter 4D-BIM definiert wurde, ermöglicht es, den aktuellen Stand des Bauvorhabens bis auf Bauteilebene abzubilden. So wird durch die Verknüpfung vom digitalen Bauwerksmodell mit der Terminplanung mittels eines Balken- oder Gantt-Diagramms ein visuell unterstützter Bauprozess ermöglicht. Zudem können im IFC-Datenschema Informationen zum Baubeginn, zum Bauende, zur verbleibenden Zeit bis zur Fertigstellung sowie zum Fertigstellungsgrad abgebildet werden.

In der Praxis führen unterschiedliche Einschätzungen des Fertigstellungsgrades in Prozent häufig zu ungenauen Ergebnissen. Daher werden Bauteile in heutigen Bauprozessen meist nur den Kategorien «in Bearbeitung» oder «fertig» zugeordnet ([7], Fokusgruppe, 2025a). Mit Blick auf die Verknüpfung mit Kostenkontrollen und Bauabnahmen könnte jedoch eine genauere Baufortschritts-Klassifizierung der Bauteile erforderlich sein. Ein möglicher Ansatz wäre zudem, Terminprogramme durch eine detailliertere IFC-Fortschrittsdarstellung zu ersetzen ([7], Fokusgruppe, 2025a). Die Experten sind sich aber einig, dass der Einbezug der IFC-Daten auf der Baustelle zukünftig eine sehr grosse Bedeutung haben wird ([3], Fokusgruppe, 2025b). Es stellt sich aber auch zukünftig die Frage, wie die Granularität des Bauplans sein soll. Damit alle Projektbeteiligten die IFC-Daten bearbeiten können, ist ein zentraler Speicherort für die Datenbank zwingend notwendig ([1], Fokusgruppe, 2025a). Selbes gilt auch für eine stabile Internetverbindung auf den Baustellen, ansonsten sind arbeiten mit Cloud-Diensten nicht möglich (Schweizer, 2025).

Der Ansatz die Verknüpfung des Baujournals mit einer Anwendung zur Abbildung des Baufortschritts in Kombination mit einem täglich genutzten Instruments ([3], Fokusgruppe, 2025b). Dieser Ansatz kann die Bauleitenden bei der Erfüllung ihrer rechtlichen Elemente aus dem Pflichtenheft gut unterstützen. Somit stellt die Applikation kein zusätzliches Tool dar, sondern wird in den Arbeitsalltag integriert und ergänzt die Baufortschrittskontrolle um normative Vorgaben der SIA (SIA 102, 2020, p. 29).

Die Hauptakteure dieser Applikation unterscheiden sich in die Nutzenden und Entwickelnden. Die Entwickelnden einer solchen Applikation werden in der Literatur auch gewünschte Nutzer benannt (Noack & Fedtke, 2025). Aus den Gesprächen mit Expert:innen aus der Branche geht hervor, dass Automatisierungen vorwiegend auf der gleichen Stufe entwickelt und angewendet werden (Bosshart, 2025; Schweizer, 2025). Allerdings sind sie keine IT-Spezialisten, welche Low-Code als schnelleres Werkzeug verwenden, sondern Fachexperten, welche ihre eigenen Anwendungen programmieren (Noack & Fedtke, 2025).

Bei den Low-Code-Plattformen gibt es Ansätze mit Model Driven Engineering (MDE), womit Applikationen beispielsweise mithilfe eines UML-Diagramms beschrieben werden könnten, anstatt dies zuerst grafisch und mit dem Layout im Frontend zu tun (France & Rumpe, 2007, p. 37). Hinsichtlich der Kompetenzen der Fachexperten aus den Interviews sind die Kompetenzen zur Beschreibung eines solchen Problems auf der Semantischen Ebene nur wenig vertreten (Schweizer, 2025). Auch nach dieser Forschung stellen sich deshalb die folgenden Fragen:

- Ob eine Befähigung der Citizen Developer zur semantischen Beschreibung einen Prototyp mit Low-Code ersetzen könnte?
- Und, könnte für komplexe Problemstellungen aus der Baubranche eine solche semantische Beschreibung mit anschliessend traditionell programmierter Applikation angestrebt werden?

6.3 Chancen und Herausforderungen

Der Low-Code-Ansatz per Drag-and-Drop um kleine Automatisierungen auf die eigenen Wünsche zusammenstellen zu können, wird als grosse Erleichterung erkannt (Hirzel, 2023; Revell, 2019, p. 1). Da die Entwicklung meist unternehmensintern erfolgt, bleiben die internen, teils vertraulichen Daten in der Unternehmung und verlassen diese nicht (Sanchis et al., 2019, p. 2). Die neue Technologie Low-Code stellt ein grosser Hoffnungsträger dar. Er verspricht, dass Fachexperten ihre eigenen Anwendungen zusammenstellen können (Shah, 2020). Womit Arbeitsgänge automatisiert und die Produktivität gesteigert werden kann (Vincent et al., 2020, p. 1). Die Erstellung von digitalen Lösungen mit Low-Code entlastet die IT-Abteilung (Mammadova et al., 2018). Häufig sind Low-Code-

Anwender in den Unternehmen eine Erweiterung zur IT-Abteilung (Blessing & Asairatnam, 2025; Bosshart, 2025), welche Low-Code-Applikationen zur Erstellung von schnellen Prototypen nutzen ([1], Fokusgruppe, 2025a; Sanchis et al., 2019, p. 3; Vesterlund & Falk, 2025). Später sollen daraus traditionell codierte Lösungen erstellt werden (Vesterlund & Falk, 2025). Bei solchen Anwendungsablösungen von Low-Code hin zu Hard-Code durch IT-Spezialisten entstehen immer wieder personelle Herausforderungen. Die Erstellenden der Low-Code-Applikationen haben oft kein Verständnis für solche weiteren Anpassungen und werden vielerorts leider als Verhinderer vom nächsten Digitalisierungsschritt gesehen ([3], Fokusgruppe, 2025a).

Allgemein lässt sich aber sagen, dass eine Reduktion der manuellen Arbeit zwingend notwendig ist, um Übertragungsfehler zu reduzieren oder gar zu verhindern (Vikebø & Sydvoid, 2019, p. 2). Neben der Zeitersparnis ist die Steigerung der Qualität der Daten der Antrieb zur Verwendung von Low-Code. Dabei lässt sich die erhoffte Erhöhung der Datenqualität aus mehreren Blickwinkeln betrachten. Einerseits werden mit der Reduktion von manueller Bearbeitung die Fehler reduziert und somit die Qualität gesteigert (Schweizer, 2025). Andererseits bleibt die Datenqualität weiterhin ein grosses Thema. Die Data Governance und die damit verbundenen Berechtigungen der Benutzer haben einen entscheidenden Einfluss auf die Datenqualität und ihre Lebensdauer (Pfister, 2024). So können Daten beispielsweise nur dann für Kalkulationen oder für Fertigung und Montage verwendet werden, wenn sie konsistent sind und die richtige Detaillierung aufweisen (Oberholzer, 2025). Stimmt die Granularität der gelieferten Daten nicht mit den geforderten Daten überein, trifft nicht nur keine erwartete Zeitreduktion ein, sondern es entsteht zusätzlicher Aufwand. In diesem Thema gehen die Firmen verschiedene Wege: Bei den einen sind ausschliesslich IT-nahe Teams für die Erstellung der Applikationen zuständig (Blessing & Asairatnam, 2025; Bosshart, 2025), während bei andern Firmen konsequent auf die Schulung der Mitarbeiter gesetzt wird (Blessing & Asairatnam, 2025; Pfister, 2024).

Es ist zwingend nötig, sich Gedanken über Prozesse und Standardisierung zu machen, denn diese sind der Grundstein für eine nachhaltige Digitalisierung der Branche. Nur wenn Prozesse vollständig neugestaltet werden, schaffen sie genug Mehrwerte (Metrolho et al., 2020, p. 8601). Trotzdem sind die Prozesse dynamisch und werden sich in Zukunft noch stärker den Bedürfnissen anpassen und agilen Führungsstilen gerecht werden müssen (Schweizer, 2025). Der aktuelle Führungsstil fordert eine hohe Wirtschaftlichkeit des Unternehmens, weil die Kultur des Niedrigsten Preises durch die Kunden gefordert wird, welche nur kurzfristige und markorientierte Beziehungen suchen (Linderoth et al., 2018). Investitionen werden daher sehr bedacht und nur dort getätigt werden, wo die grössten Kostenersparnisse zu erwarten sind (Vesterlund & Falk, 2025). Zudem soll der Arbeitsinhalt so optimiert werden, dass überwiegend wertschöpfende Arbeit und nur wenig nicht wertschöpfende verrichtet wird (Pfister, 2024). Dabei bleiben die Dienstleister meist auf den Investitionskosten für die Softwares sitzen (Vesterlund & Falk, 2025). Auch der Aufwand und die Kosten für Schulungen sind kostenintensiv (Blessing & Asairatnam, 2025).

Trotzdem besteht das Problem, dass Unternehmungen hardcodierte Codes in die missliche Lage bringen und dass diese bei Abwesenheit der Erstellenden trotz Dokumentation oft nicht mehr nachvollziehbar sind (Blessing & Asairatnam, 2025). In Bezug auf Low-Code widerspricht die Literatur der Realität. So beschreiben (Noack und Fedtke) Low-Code als Lösung, welche eine Dokumentation überflüssig macht, im Gegenzug spricht die Fokusgruppe von fehlender Nachvollziehbarkeit in Abwesenheit des Erstellers ([3], Fokusgruppe, 2025a). Allerdings bietet Autodesk bereits ein ACC-Tool für den Baufortschritt an. Insgesamt bleibt es eine Frage der budgetierten Lizenzkosten im Vergleich mit den Entwicklungs- und Mehrwertkosten einer eigens entwickelten Low-Code-Applikation.

6.4 Konkrete Anwendungsszenarien

Allgemein wird Low-Code in der Baubranche vermehrt für interne Prozessautomationen oder -optimierungen angewendet ([1], Fokusgruppe, 2025a). Dies sind häufig Listen, welche in einer Cloud gespeichert sind und über Applikationen bearbeitet werden. Ein gutes Beispiel dazu ist die Applikation zur Dokumentation des Sicherheitsrundgangs (Schweizer, 2025). Anstatt nach dem Rundgang eine Excel-Liste im Büro auszufüllen, kann während des Rundgangs eine Checkliste auf einer Low-Code-Applikation abgearbeitet werden, die jederzeit ohne Nachbearbeitung wieder hervorgeholt werden kann. Auch die automatisierte Datenverschiebung ins und aus dem Archiv kann als gängiges und sinnvolles Beispiele für die Verwendung von Low-Code angesehen werden (Blessing & Asairatnam, 2025). Gerne wird Low-Code für persönliche Automatisierungen verwendet. Zudem macht die Nutzung unternehmensübergreifender Applikationen immer wieder Probleme ([8], Fokusgruppe, 2025a). In anderen Branchen wird Low-Code beispielsweise auch für die Fabrikation eingesetzt (UiPath Inc, 2025), wovon in der Baubranche eher abgesehen werden kann. Jedes Projekt ist wieder anders und hat andere Anforderungen

([7], Fokusgruppe, 2025a). Deshalb wird Low-Code nur bedingt zur projektübergreifenden Anwendung eingesetzt, da eine Vereinheitlichung fehlt. Vielmehr müsste die Schweiz von ihren eigenen Standards wie SIA, CRB und e-BKP absehen um länderübergreifend gemeinsame Standards und das entsprechende Wissen nutzen zu können (Oberholzer, 2025).

Auffällig ist, dass in der Baubranche viele Anwendungsfälle aus der Praxis, auf die Kostenkontrolle abzielen (Bossart, 2025; Schweizer, 2025; Vesterlund & Falk, 2025). Beliebte sind deshalb auch Dashboards, mit welchen sich Projekte anhand immer gleicher Aspekte vergleichen lassen. Die Anforderungen kommen in einigen Firmen direkt von der Baustelle und werden für die ganze Unternehmung erstellt (Bossart, 2025; Schweizer, 2025). Meist erstellen die Anwendenden die Automatisierungen aber primär für sich und seine Zwecke selbst (Pfister, 2024). Nur in vereinzelt Unternehmungen sind die Prozesse so standardisiert, dass die Applikationen für die ganze Belegschaft ausgelegt werden können (Blessing & Asairatnam, 2025). Werden branchenspezifischen Automatisierungen angestrebt, führt dies zu Problemen, da sich die Anwendungen an interne Standards, sowie Schnittstellen und Compliance-Richtlinien halten müssen (Pfister, 2024).

Die Baubranche in der Schweiz ist allgemein eher träge. Daher ist es wenig überraschend, dass eine Veränderung immer viel Zeit und mutige Early Adopters voraussetzen (Bossart, 2025). Dabei muss eine Applikation immer von der Unternehmung getragen werden können, in der sie entsteht und verwendet wird ([3], Fokusgruppe, 2025a). So braucht das Change-Management auch bei dieser Technologie auch immer die geeignete organisatorische Umstellung, um die digitalisierten Prozesse zu verändern. Der wesentliche Antrieb sollte dabei immer beim Mehrwert der Digitalisierung liegen, der dem Kunden eine bessere Erfahrung oder eine Zeit-/Kostensparnis bieten soll (Vesterlund & Falk, 2025).

6.5 Microsoft Power Platform vs. Autodesk Platform Services

Die zunehmende Digitalisierung der Bauprozesse führt zu einer wachsenden Bedeutung von Plattformlösungen, welche den Umgang mit IFC-Daten ermöglichen. Besonders in den Phasen der Planung und Bauausführung ist die effiziente Nutzung und Manipulation von IFC-Daten sehr wichtig, um eine konsistente Datenbasis über den gesamten Lebenszyklus eines Bauwerks sicherzustellen. Die beiden Plattformansätze stehen dabei exemplarisch für unterschiedliche Paradigmen. Die Microsoft Power Platform setzt auf der einen Seite auf Low-Code-Prinzipien für Citizen Developer (Rymer & Koplowitz, 2019). Die Autodesk Platform Services ist auf der anderen Seite als spezialisierte BIM-Cloud-Dienste konzipiert, welche auch die Entwicklung von Low-Code-Applikationen ermöglicht (Autodesk Inc., 2025b, 2024). In dieser Studie wurde primär die Microsoft Power Platform für die Erstellung des Prototyps genutzt, da diese Plattform gemäss Aussagen in den Interviews am meisten verwendet wird (Siehe Abbildung 37).

Ein sehr wichtiger Aspekt ist der Umgang mit Fehlerbehebungen beim Aufbau und Betrieb solcher Low-Code-Applikationen. Wie Woo beschreibt, darf dieser Aspekt während des Erstellungsprozesses nicht unterschätzt werden (Woo, 2020). Gerade bei der Microsoft Power Platform wird deutlich, dass Citizen für die Erstellung von Applikationen viel Zeit benötigen. Sie sind dabei bei anfallenden Problemen häufig auf die Unterstützung durch den IT-Support angewiesen (Blessing & Asairatnam, 2025). Im Zusammenhang mit dem Prototyp dieser Thesis kann diese Ansicht bestätigt werden. So waren stetig ändernde Datenschutzrichtlinien vom Hochschulaccount dafür verantwortlich, dass Schnittstellen zu den Quelldaten aus der Datenbank schnell Fehler aufweisen und stetig wieder überarbeitet werden oder die Zugriffsrechte immer wieder neu vergeben mussten. Dieses Problem verstärkt sich in der Planungsphase, wenn kurzfristige Plananpassungen erforderlich sind, und wirkt sich in der Ausführungsphase auf der Baustelle noch kritischer aus, da hier ausschliesslich die gültigen Daten aus dem IFC gefordert sind (Schweizer, 2025).

Allgemein muss sich Low-Code immer die Frage gefallen lassen, bis zu welchem Punkt eine Ausarbeitung sinnvoll ist und ab wann eine traditionelle Entwicklersprache bevorzugt wird ([1], Fokusgruppe, 2025a). Während einfache Automatisierungen von Arbeitsabläufen in der Planung (z.B. das automatische Ableiten von Dashboards aus Terminplänen) durchaus realisierbar sind, stossen Citizen Developer bei der Arbeit mit IFC-Daten schnell an Grenzen. Wie vorangehend beschrieben erfordert IFC tiefere Kenntnisse in Datenmodellierung und Strukturlogik, die über die in der Power Platform übliche Low-Code/No-Code-Philosophie hinausgehen. Hier zeigt sich der Nachteil, dass für die Power Platform keine traditionelle Entwicklersprachen wie Python oder C# erforderlich ist. Dadurch ist es jedoch schwieriger, komplexere Logik umzusetzen ([3],[1], Fokusgruppe, 2025a). Für projektübergreifende Anwendungen wie die modellbasierte Terminplanung, die im Prototyp umgesetzt wurde, sind in der Regel

professionelle Entwickler erforderlich. Dies widerspricht jedoch der ursprünglichen Idee des Citizen Developments. Sobald Anwendungen projektübergreifend eingesetzt werden, können Lizenzkosten einen entscheidenden Einfluss auf den Einsatz der gewählten Technologie haben ([1], Fokusgruppe, 2025a). So wurde im Rahmen dieser Thesis die im Softwareplan vorgesehene maximale Anzahl von drei Applikationen bereits erreicht, sodass keine weiteren mehr erstellt werden können. Autodesk Platform Services hingegen setzen auf ein nutzerbasiertes Modell, wobei die Nutzenden ihre Lizenz dabei auch für mehrere Projekte nutzen können. In grossen Projekten mit vielen externen Beteiligten ist das oft flexibler und wirtschaftlicher da viele Unternehmen bereits eigene solche Lizenzen haben (Blessing & Asairatnam, 2025).

Die Weiterverarbeitung von IFC ist mit der Microsoft Power Platform nicht vorgesehen. Deshalb können die Daten nur eingesehen, aber nicht bearbeitet werden und die Datei kann nur mit einem hardcodierten Code weiterverarbeitet werden. Dies stellt im Zusammenhang mit der angedachten Zielgruppe der Entwickler ein wesentliches Problem dar. Citizen Developer sind Fachpersonen in der Baubranche, die in der Regel keine Codes in den traditionellen Entwicklersprachen schreiben können. Ausserdem wird sowohl in dieser Thesis als auch von der Fokusgruppe die Erkenntnis gewonnen, dass diese Technologie schwierig zu handhaben ist, sobald mehr als einfache Datenabfragen durchgeführt werden und man bei einer Enterprise-Applikation ankommt, und die Erstellungskosten nicht mehr im Verhältnis zum Nutzen stehen([3],[1], Fokusgruppe, 2025a). Einige Interviewpartner bestätigen, dass die Kombination von Microsoft PowerApps und IFC sehr schwierig ist (Schweizer, 2025). Da die Schnittstellen sehr dynamisch sind, können Datenverbindungen aufgrund eines Authentifizierungsfehlers nicht mehr funktionieren. Es ist deshalb zu überlegen, ob andere Hersteller bei Projekten mit IFC Abhilfe bieten können.

Aufgrund dieser Einschränkungen wurde eine andere Low-Code-Plattform gesucht, welche IFC unterstützt. Ein vielversprechender Ansatz bietet hier Model Coordination tool,. Ähnlich wie bei Microsoft steht darin eine Cloud zur Speicherung von Dokumenten sowie eine direkte Anbindung einer App zur Verfügung, wodurch sich ebenfalls ein Ökosystem ergibt. Der grosse und wesentliche Unterschied liegt jedoch darin, dass Autodesk für die Bearbeitung von Bauteilen konzipiert ist und die Dokumentart IFC somit eine bekannte Datenstruktur ist. Dadurch wird ein direkter Zugriff auf die Metadaten von IFC sowie eine interaktive Darstellung des 2D/3D-Modells in der Mobile-App ermöglicht. In der Fokusgruppe ist man sich einig, dass ein Viewer nötig ist, um sich zurechtzufinden ([1], Fokusgruppe, 2025a). Am Ende muss die eingesetzte Technologie, egal ob über Low-Code-Applikationen oder das Model Coordination tool, die Projektteams bestmöglich in ihren Tätigkeiten unterstützen.

7 Fazit

Die Dokumentation des Baufortschritts ist einer der wesentlichen Bestandteil jeder operativen Bauführung. In der Praxis erfolgt sie jedoch häufig manuell, unstrukturiert und medienbruchbehaftet, beispielsweise durch handschriftliche Notizen, Excel-Listen oder nicht verknüpfte Fotodokumentationen. Dies führt zu Inkonsistenzen, einem hohen administrativen Aufwand und einer eingeschränkten Nachvollziehbarkeit. Um diesem Problem zu begegnen, wird in dieser Arbeit ein digital unterstützter Prozess zur Fortschrittsdokumentation auf Basis des IFC-Datenmodells in Verbindung mit der Low-Code-Technologie vorgestellt. Dabei wurde als spezifischer Anwendungsfall die Integration des digitalen Baujournals in den Arbeitsalltag der Bauleitung beleuchtet.

Das IFC-Datenformat ermöglicht die bauteilgenaue und standardisierte Abbildung von Bauwerksinformationen inklusive Geometrie-, Zeit- und Statusinformationen. In Kombination mit einer Low-Code-Plattform kann ein digitales Baujournal entwickelt werden, das auf diese strukturierten Daten zugreift und sie benutzerfreundlich visualisiert. So lässt sich beispielsweise der Status einzelner Bauteile direkt in Prozent über eine mobile App erfassen und mit ergänzenden Informationen wie Mängeln, Witterungseinflüssen oder Fotodokumentationen verknüpfen. Der Fortschritt kann automatisch mit Zeitplänen oder Balkendiagrammen abgeglichen werden.

Der grosse Vorteil von Low-Code besteht darin, dass Fachpersonen (Citizen Developer) die Applikationen ohne tiefe Programmierkenntnisse selbst erstellen und anpassen können. So können Bauleiter:innen eigene Workflows zur Fortschrittskontrolle modellieren, automatisierte Erinnerungen oder Eskalationen bei Terminverzug einrichten und Dashboards zur Baustellenüberwachung konfigurieren. Durch die Verknüpfung mit SharePoint oder anderen Cloud-Systemen ist eine zentrale, versionierte Ablage aller Journaleinträge möglich, auf die alle Projektbeteiligten zugreifen können. Meist kommen Low-Code-Anwendungen für interne Prozesse wie automatisierte Mail-Antworten, bei erfolgreichem Hochladen einer Datei oder Listen bearbeiten über eine Mobile-Applikation. Diese Anwendungen werden bereits häufig angewendet und haben darin auch ihre Sinnhaftigkeit. Ebenfalls wird die

Low-Code-Technologie häufig für Firmeninterne Prozesse genutzt. Auch diese Arbeitsweise macht im Zusammenhang mit Datenschutzrichtlinien Sinn.

In einem prototypischen Anwendungsfall wurde eine Low-Code-Applikation entwickelt. Sie verbindet ein digitales Baujournal mit einem digitalen Bauwerksmodell auf der Basis von IFC-Daten. Die Bauleitung kann damit den Fortschritt direkt auf der Baustelle per Tablet festhalten. Dies erfolgt bauteilbezogen, visuell und zeitsynchron. So erfüllt das Baujournal die normativen Anforderungen gemäss SIA 102 und wird zugleich zu einem interaktiven Werkzeug im Tagesgeschäft.

Eine Herausforderung bei der technischen Umsetzung besteht vor allem in der strukturierten Extraktion und Rückführung der IFC-Metadaten. Während sich das Lesen und Anzeigen dieser Informationen in Low-Code-Apps gut umsetzen lässt, ist dies beim Zurückschreiben von neuen Informationen deutlich komplexer. Anpassungen wie der Fertigstellungsgrad erfordern komplexere Datenflüsse oder gegebenenfalls die Einbindung von API oder Zwischenformaten wie beispielsweise CSV-Dateien. Trotzdem ist klar, dass die digitale Erfassung eine Erhöhung der Datenqualität mit sich bringt und gleichzeitig den manuellen Aufwand senkt und die Zusammenarbeit auf der Baustelle verbessert werden.

Die Kombination von Low-Code-Technologie mit IFC basierenden Daten stellt eine vielversprechende Lösung zur Digitalisierung des Baujournals und der Fortschrittsdokumentation dar. Durch die einfache Anpassbarkeit und die visuelle Interaktion mit dem 3D-Modell durch die Applikations-Bedienenden wird der Baufortschritt nicht nur dokumentiert, sondern aktiv gesteuert. Damit wird ein wichtiger Schritt in Richtung durchgängig digitaler Bauprozesse realisiert, der nah an der Praxis ist und den Fokus auf Effizienz, Nachvollziehbarkeit und Datenqualität legt. Trotz aller positiven Tendenzen hat die Lösung der Schnittstellenproblematiken zwischen den unterschiedlichen Systemen und Applikationen den grössten Einfluss auf die Zusammenarbeit von digitalen Bauwerksmodellen und Low-Code-Applikationen.

7.1 Schlussfolgerung

In dieser Masterarbeit wurde ein Ansatz zur Integration von digitalen Bauwerksmodellen auf der Basis vom IFC-Datenschema in Low-Code-Technologie theoretisch hergeleitet und entwickelt. Weiter wurde dieser anhand eines exemplarischen Praxisbeispiels umgesetzt, womit die zentrale Forschungsfrage der Arbeit:

Wie könnte die Low-Code Technologie im Zusammenhang mit IFC und dem digitale Bauwerksmodelle in den Arbeitsalltag integriert werden?

aus verschiedenen Blickwinkeln betrachtet und auf wissenschaftlicher Ebene beantwortet werden.

Mit der Abhandlung konnten zudem Antworten auf die Teilfragen der theoretischen Variantenbildung sowie der technischen Umsetzung gegeben werden. Dazu wurden die Mehrwertpotenziale durch den Einsatz von Low-Code sowie die Chancen und Herausforderungen beim Einsatz in der Baubranche aufgezeigt werden. Es wurde ein Prototyp erstellt, der exemplarisch das Bauteil Wand aufzeigt und veranschaulicht, wie der Baufortschritt in einem IFC in der Praxis angewendet werden könnte und wie dies im Instrument des Baujournals zum Einsatz kommen könnte.

7.2 Ausblick

Der in dieser Master-Thesis entwickelte Workflow zur Messung und beschreiben eines Baufortschrittes im IFC-Datenformat bezieht sich ausschliesslich auf die Ausführungsphase, in der Schweiz und wurde nur modellhaft für das Bauteil Wand getestet. In einem nächsten Schritt ist für die praktische Umsetzung die Durchführung eines entsprechenden Pilotprojekts mit realen Bauvorhaben erforderlich. In dieses sind alle relevanten Akteure der Bauleitung und Planer aktiv eingebunden und die Detaillierung des Baufortschritts ist geklärt. Dadurch könnten praxisnahe Erkenntnisse über Anwendbarkeit, Nutzerakzeptanz und technische Integration gewonnen werden.

Diese Erkenntnisse bilden die Grundlage für eine kontinuierliche Weiterentwicklung und Verfeinerung des Workflows. Das Ziel besteht darin, eine Lösung zu schaffen, die nicht nur für Einzelprojekte geeignet ist, sondern auch auf unterschiedliche Gebäudearten und Projektgrössen skalierbar ist. Ebenfalls ist zu klären wie der Baufortschritt in den Bauteilen abgebildet wird und in welcher Detaillierung.

Während der Untersuchung wurde festgestellt, dass die Aktualisierung der Datenbank und die damit verbundenen Freigabenzyklen ein entscheidendes Thema ist, welches in diesem Umfang nicht vertieft bearbeitet wurde.

Ebenfalls wurde der Prototyp mit der Microsoft-Power Platform umgesetzt. Um ganzheitliche Erkenntnisse zu erlangen, müssten auch andere Plattformen getestet werden. Auch gibt es viele Ansätze in welchen Baujournals mit Fotos ergänzt werden können. Auch dieser Ansatz müsste gegeben falls integriert werden.

Aus Sicht der Verfasserin ist unklar, ob die digitale Transformation des Baujournals mittels Low-Code die Sinnhaftigkeit einer solchen Applikation gewährleistet. Die Kombination von Low-Code und IFC ist aktuell nicht für eigene Apps mit der Microsoft Power Platform möglich. Auch wenn damit individuelle Aspekte berücksichtigt werden können und sensible Daten in der Unternehmung bleiben, macht bei einer solch komplexen Datenverarbeitung eine fertige Lösung eines Software-Herstellers Sinn. Die kürzere Entwicklungszeit einer Applikation trifft nur bei überschaubaren Applikationen, ein.

Für künftige Forschung sollte geprüft werden, ob der Fokus bei den Kompetenzen der Fachexperten darauf liegen sollte, Anforderungen semantisch beschreiben zu können, um sie anschliessend von IT-Spezialisten umsetzen zu lassen. So soll im Zusammenhang mit IFC und Low-Code der Fokus auf den Schnittstelle liegen.

8 Quellenverzeichnis

- Adrian, B., Hinrichsen, S., Nikolenko, A., 2020. *App Development via Low-Code Programming as Part of Modern Industrial Engineering Education*. Adv. Hum. Factors Syst. Interact. 45–51. https://doi.org/10.1007/978-3-030-51369-6_7
- Aftab, O., Al, V.P. of C., 2023. Microsoft Power Virtual Agents, now part of Microsoft Copilot Studio [WWW Document]. Microsoft Copilot Blog. URL <https://www.microsoft.com/en-us/microsoft-copilot/blog/copilot-studio/microsoft-power-virtual-agents-now-part-of-microsoft-copilot-studio/> (accessed 7.23.25).
- Alsaadi, H.A., Radain, D.T., Alzahrani, M.M., Alshammari, W.F., Alahmadi, D., Fakieh, B., 2021. *Factors that affect the utilization of low-code development platforms: survey study*. Rev. Rom. Informatică Și Autom. 31, 123–140. <https://doi.org/10.33436/v31i3y202110>
- Ameller, D., Burgués, X., Collell, O., Costal, D., Franch, X., Papazoglou, M.P., 2015. *Development of service-oriented architectures using model-driven development: A mapping study*. Inf. Softw. Technol. 62, 42–66. <https://doi.org/10.1016/j.infsof.2015.02.006>
- AppSheet, 2019. Enabling Digital Transformation in the Construction Industry [WWW Document]. URL <https://cloud.google.com/blog/products/no-code-development?hl=en>
- Autodesk Inc., 2025a. BUILD Hilfe | Verwalten von Formularvorlagen | Autodesk [WWW Document]. URL https://help.autodesk.com/view/BUILD/DEU/?guid=Build_Forms_Manage (accessed 9.15.25).
- Autodesk Inc., 2025b. Autodesk Platform Services (formerly Forge) [WWW Document]. URL <https://aps.autodesk.com/> (accessed 9.15.25).
- Autodesk Inc., 2024. Autodesk Construction Cloud APIs & Integrations | Autodesk Platform Services [WWW Document]. URL <https://aps.autodesk.com/autodesk-construction-cloud-apis-integrations> (accessed 9.15.25).
- Autodesk Platform Services, 2025. Announcing Data Exchange for IFC Public Beta [WWW Document]. URL <https://aps.autodesk.com/blog/announcing-data-exchange-ifc-public-beta> (accessed 8.5.25).
- Azhar, S., Ahmad, I., Sein, M.K., 2009. *Action Research as a Proactive Research Method for Construction Engineering and Management*, American Society of Civil Engineers. J. Constr. Eng. Manag. 136, 87–98. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000081](https://doi.org/10.1061/(ASCE)CO.1943-7862.0000081)
- Ballard, G., 2000. The Last Planner System of Production Control. University of Birmingham, Birmingham.
- Becker, J., Delfmann, P., 2007. *Reference modeling: Efficient information systems design through reuse of information models*. Physica-Verlag, Münster.
- Bitkom, 2018. 82.000 freie Jobs: IT-Fachkräftemangel spitzt sich zu | Presseinformation | Bitkom e. V. [WWW Document]. URL <https://www.bitkom.org/Presse/Presseinformation/82000-freie-Jobs-IT-Fachkraefte-mangel-spitzt-sich-zu> (accessed 10.13.24).
- Blessing, M., Asairatnam, M., 2025. *Expertengespräch: Anwendung Low-Code Technologie beim Elektrotechnik-Unternehmen*. Münchenstein. Geführt von Leah Lüscher am 07.04.2025.
- Bock, A., Frank, U., 2021. *Low-Code Platform*. Bus. Inf. Syst. Eng. 63, 733–740. <https://doi.org/10.1007/s12599-021-00726-8>
- Borrmann, A., König, M., Koch, C., Beetz, J. (Eds.), 2021. *Building Information Modeling: Technologische Grundlagen und industrielle Praxis*, VDI-Buch. Springer Fachmedien Wiesbaden, Wiesbaden. <https://doi.org/10.1007/978-3-658-33361-4>
- Bosshart, N., 2025. *Expertengespräch: Anwendung Low-Code Technologie beim Gleisbau-Unternehmen*. Bern. Geführt von Leah Lüscher am 26.03.2025.
- buildagil, 2025. Datenanalyse mit Power Query [WWW Document]. Knowl. Base Buildagil. URL <https://knowledge.buildagil.com/de/article/datenanalyse-mit-power-query-jp7b6e/> (accessed 8.31.25).

- buildingSMART International, 2024a. Annex B (informative) Alphabetical listings - IFC 4.3.2 Documentation [WWW Document]. URL <https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/annex-b.html> (accessed 9.12.25).
- buildingSMART International, 2024b. 5.3 IfcProcessExtension - IFC 4.3.2 Documentation [WWW Document]. URL <https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/ifcprocessextension/content.html> (accessed 9.12.25).
- buildingSMART International, 2024c. Annex E (informative) Examples - IFC 4.3.2 Documentation [WWW Document]. URL https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/annex_e/construction-scheduling/construction-scheduling-task.html (accessed 7.10.25).
- buildingSMART International, 2024d. 8.5.3.9 IfcTaskTime - IFC 4.3.2 Documentation [WWW Document]. URL <https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/ifcTaskTime.htm> (accessed 7.8.25).
- Makler- und Bauträgerverordnung - MaBV, 2023. *Verordnung über die Pflichten der Immobilienmakler, Darlehensvermittler, Bauträger, Baubetreuer und Wohnimmobilienverwalter (Makler- und Bauträgerverordnung - MaBV)*.
- Chang, Y.-H., Ko, C.-B., 2017. *A Study on the Design of Low-Code and No Code Platform for Mobile Application Development*, The Institute of Internet, Broadcasting and Communication. Int. J. Adv. Smart Converg. 6, 50–55. <https://doi.org/10.7236/IJASC.2017.6.4.7>
- Christensen, B. H., 2018. *Anskaffelse og Implementering av forretningssystemer: Forretningsutvikling og gevinstrealisering i en digital tidsalder*.
- Chuong Mep, 2024. How To Export IFC Data To Excel Or CSV [WWW Document]. URL <https://chuongmep.com/posts/2024-02-25-Export-IFC-Excel-Csv.html> (accessed 8.2.25).
- Costagliola, G., Deufemia, V., Polese, G., 2004. *A framework for modeling and implementing visual notations with applications to software engineering*. ACM Trans. Softw. Eng. Methodol. 13, 431–487. <https://doi.org/10.1145/1040291.1040293>
- DiCicco-Bloom, B., Crabtree, B.F., 2006. *The qualitative research interview*. Med. Educ. 40, 314–321. <https://doi.org/10.1111/j.1365-2929.2006.02418.x>
- Duden, 2025. Bauleitung: Rechtschreibung, Bedeutung, Definition, Herkunft [WWW Document]. URL <https://www.duden.de/rechtschreibung/Bauleitung> (accessed 7.9.25).
- Elzarka, H., 2017. *Computer integrated construction for small and medium contractors*. Univ. Cincinnati Cincinnati Ohio.
- Fiedler, M., 2018. *Lean Construction – Das Managementhandbuch Agile Methoden und Lean Management im Bauwesen*. Springer Gabler, München.
- Fokusgruppe, 2025a. *Fokusgruppe Praxispartner zur Besprechung des Prototyps*. Opfikon. Geführt von Leah Lüscher am 12.08.2025.
- Fokusgruppe, 2025b. *Nutzwertanalyse zur Bewertung des Prototyps durch den Praxispartner*. Opfikon. Geführt von Leah Lüscher am 12.08.2025.
- fortune business insights, 2024. Low Code Development Platform Market Size, Share [2024 - 2032] [WWW Document]. URL <https://www.fortunebusinessinsights.com/low-code-development-platform-market-102972> (accessed 9.19.24).
- France, R., Rumpe, B., 2007. *Model-driven Development of Complex Software: A Research Roadmap*. Presented at the Proc. of the 29th International Conference on Software Engineering, Future of Software Engineering Trac, pp. 37–54. <https://doi.org/10.1109/FOSE.2007.14>
- Frank, U., 2013. *Domain-Specific Modeling Languages: Requirements Analysis and Design Guidelines*, Springer. Springer, pp. 133–157. https://doi.org/10.1007/978-3-642-36654-3_6

- Gann, D.M., 1996. *Construction as a manufacturing process? Similarities and differences between industrialized housing and car production in Japan*. *Constr. Manag. Econ.* 14, 437–450. <https://doi.org/10.1080/014461996373304>
- Gantt, H.L., 1910. *Work, wages, and profits: their influence on the cost of living*, Works management library. Engineering magazine, New York.
- Google, 2025. Blockly [WWW Document]. Google Dev. URL <https://developers.google.com/blockly?hl=de> (accessed 8.9.25).
- Haghsheno, S., Wachter, N., 2019. *Lean Construction - Begriffe und Methoden*, 1. Auflage. ed. German LEAN Construction Institut, Karlsruhe.
- Harty, C., 2005. *Innovation in construction: a sociology of technology approach*, Routledge. *Build. Res. Inf.* 33, 512–522. <https://doi.org/10.1080/09613210500288605>
- Hirzel, M., 2023. *Low-Code Programming Models - Low-code has the potential to empower more people to automate tasks by creating computer programs*. *Commun. ACM* 66, 76–85. <https://doi.org/10.1145/3587691>
- IBM, 2022. Maximo BIM Forge Viewer PlugIn [WWW Document]. URL <https://www.ibm.com/support/pages/maximo-bim-forge-viewer-plugin> (accessed 8.5.25).
- IBM, 2021. What Is Low-Code? | IBM [WWW Document]. What Low-Code. URL <https://www.ibm.com/topics/low-code> (accessed 8.4.24).
- IfcOpenShell Contributors, 2025. ifcopenshell - IfcOpenShell 0.8.3 documentation [WWW Document]. URL <https://docs.ifcopenshell.org/autoapi/ifcopenshell/index.html#module-ifcopenshell> (accessed 8.2.25).
- Ingalls, D., Wallace, S., Chow, Y.-Y., Ludolph, F., Doyle, K., 1988. *Fabrik: a visual programming environment*, Association for Computing Machinery, in: Conference Proceedings on Object-Oriented Programming Systems, Languages and Applications, OOPSLA '88. Association for Computing Machinery, New York, NY, USA, pp. 176–190. <https://doi.org/10.1145/62083.62100>
- Jacobsson, M., Linderoth, H., 2010. *The influence of contextual elements, actors' frames of reference, and technology on the adoption and use of ICT in construction projects: a Swedish case study*. *Constr. Manag. Econ.* 28, 13–23. <https://doi.org/10.1080/01446190903406154>
- Kelly, S., Tolvanen, J.-P., 2008. *Domain-Specific Modeling: Enabling Full Code Generation*. A Wiley-Interscience Publication, Hoboken, New Jersey.
- Khorrarn, F., Mottu, J.-M., Sunyé, G., 2020. *Challenges & opportunities in low-code testing*, ACM, in: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings. Presented at the MODELS '20: ACM/IEEE 23rd International Conference on Model Driven Engineering Languages and Systems, ACM, Virtual Event Canada, pp. 1–10. <https://doi.org/10.1145/3417990.3420204>
- Kurt Lewin, 1953. *Die Lösung sozialer Konflikte: ausgewählte Abhandlungen über Gruppendynamik*. Im Christian-Verl., Bad Nauheim.
- Linderoth, H.C.J., Jacobsson, M., Elbanna, A., Royal Holloway University of London, GB, 2018. *Barriers for Digital Transformation: The Role of Industry*, University of Technology, Sydney, in: Australasian Conference on Information Systems 2018. University of Technology, Sydney. <https://doi.org/10.5130/acis2018.az>
- Love, P.E.D., Gunasekaran, A., Li, H., 1998. *Concurrent engineering: a strategy for procuring construction projects*. *Int. J. Proj. Manag.* 16, 375–383. [https://doi.org/10.1016/S0263-7863\(97\)00066-5](https://doi.org/10.1016/S0263-7863(97)00066-5)
- Löwstedt, M., Räisänen, C., 2014. *Social identity in construction: enactments and outcomes*, Informa UK Limited. *Constr. Manag. Econ.* 32, 1093–1105. <https://doi.org/10.1080/01446193.2014.956132>
- Mammadova, M., Jabrayilova, Z., Mammadzada, F.R., 2018. *Intelligent management of the unbalance supply chain and demand in the labor market for IT specialists*. *Int. J. Supply Chain Manag.* 7, 347–354.
- Martinez, E., 2024a. *Startbesprechung zur Master-Thesis und aktueller Stand Implenia CH*.

- Martinez, E., 2024b. *Expertengespräch: Forschungsbedarf in der Branche mit Low-Code*.
- Martinez, E., Cisterna, D., 2023. *Using Low-Code and Artificial Intelligence to Support Continuous Improvement in the Construction Industry*. Presented at the 31st Annual Conference of the International Group for Lean Construction (IGLC 31), Lille, France, pp. 197–207. <https://doi.org/10.24928/2023/0236>
- Matvitsky, 2023. Magic Quadrant for Enterprise Low-Code Application Platforms [WWW Document]. Gartner. URL <https://www.gartner.com/en/documents/4843031> (accessed 9.13.24).
- Mayring, P., 2022. *Qualitative Inhaltsanalyse - Grundlagen und Techniken*, 13. Auflage. ed. BELTZ, Weinheim Basel.
- Menz, S., 2020. map | Bauleitung [WWW Document]. URL <https://map.arch.ethz.ch/artikel/28/bauleitung> (accessed 7.8.25).
- Messerli Informatik, 2021. BAUAD - Softwaretechnologie für die Bauplanung [WWW Document]. URL <https://www.messerli.ch/dokumente/bauad/index.html#page=1> (accessed 4.12.25).
- Metrolho, J., Araujo, R., Ribeiro, F., 2020. *A STRATEGY FOR FACING NEW EMPLOYABILITY TRENDS USING A LOWCODE DEVELOPMENT PLATFORM*. Presented at the INTED2020 Conference, Valencia, Spain. <https://doi.org/DOI:%252010.21125/inted.2020.2341>
- Microsoft, 2025a. Was ist Microsoft Dataverse? - Power Apps [WWW Document]. URL <https://learn.microsoft.com/de-de/power-apps/maker/data-platform/data-platform-intro> (accessed 7.23.25).
- Microsoft, 2025b. Power BI – Datenvisualisierung | Microsoft Power Platform [WWW Document]. URL <https://www.microsoft.com/de-de/power-platform/products/power-bi> (accessed 7.23.25).
- Microsoft, 2025c. Offizielle Microsoft Power Apps-Dokumentation - Power Apps [WWW Document]. URL <https://learn.microsoft.com/de-de/power-apps/> (accessed 7.23.25).
- Microsoft, 2025d. Microsoft Power Automate – Prozessautomatisierungsplattform | Microsoft [WWW Document]. URL <https://www.microsoft.com/de-de/power-platform/products/power-automate> (accessed 7.23.25).
- Microsoft, 2025e. Dokumentation zu Microsoft Power Pages [WWW Document]. URL <https://learn.microsoft.com/de-de/power-pages/> (accessed 7.23.25).
- Microsoft, 2025f. Übersicht über AI Builder [WWW Document]. URL <https://learn.microsoft.com/de-de/ai-builder/overview> (accessed 7.23.25).
- Moskal, M., 2021. *NO-CODE APPLICATION DEVELOPMENT ON THE EXAMPLE OF LOGOTEC APP STUDIO PLATFORM*. Inform. Autom. Pomiarowy W Gospod. Ochr. Śr. 11, 54–57. <https://doi.org/10.35784/iapgos.2429>
- Mosquera, D., Ruiz, M., Pastor, O., Spielberger, J., 2024. *Understanding the landscape of software modelling assistants for MDSE tools : a systematic mapping*. Elsevier BV This Open Access Artic. CC License <https://creativecommons.org/licenses/by/4.0/> 173. <https://doi.org/10.1016/j.infsof.2024.107492>
- Mukerji, J., Miller, J., 2003. *MDA Guide V1.0.1(omg/03-06-01), Technical report, OMG*. MDA Guide Version 101.
- Navin, M., 2023. *EMPOWERING CITIZEN DEVELOPERS: EXPLORING COPILOT'S ROLE IN LOW-CODE APP DEVELOPMENT*. Int. Res. J. Mod. Eng. Technol. Sci.
- Noack, K., Fedtke, S. (Eds.), 2025. *Paradigmenwechsel Low-Code: Softwareentwicklung (fast) ohne Programmierung*, KI - Best Practice in Industrie, Dienstleistung und Verwaltung. Springer Vieweg, Wiesbaden. <https://doi.org/10.1007/978-3-658-47126-2>
- Oberholzer, C., 2025. *Expertengespräch: Anwendung Low-Code Technologie mit einem Digital Consulting-Unternehmen*. Luzern. Geführt von Leah Lüscher am 09.04.2025.
- Osmundsen, K., Iden, J., Bygstad, B., 2018. *Hva er digitalisering, digital innovasjon og digital transformasjon?* Presented at the NOKOBIT 2018 - Norsk konferanse for bruk av InformasjonsTeknologi, Svalbard, p. 16.
- outsystems, 2019. The State of Application Development [WWW Document]. URL <https://www.outsystems.com/1/state-app-development-trends/>
- Overeem, M., Jansen, S., Mathijssen, M., 2021. *API Management Maturity of Low-Code Development Platforms*, Springer International Publishing, in: Augusto, A., Gill, A., Nurcan, S., Reinhartz-Berger, I., Schmidt, R.,

- Zdravkovic, J. (Eds.), *Enterprise, Business-Process and Information Systems Modeling*. Springer International Publishing, Cham, pp. 380–394. https://doi.org/10.1007/978-3-030-79186-5_25
- Palacios-González, E., Fernandez, H., García Díaz, V., Pelayo García-Bustelo, B., Cueva Lovelle, J., Sanjuán, O., 2008. *General purpose MDE tools*. *Int. J. Interact. Multimed. Artif. Intell.* 1, 72–75.
- Pfister, L., 2024. *Expertengespräch: Anwendung Low-Code Technologie beim Totalunternehmer*. Opfikon. Geführt von Leah Lüscher am 25.10.2024.
- proLogistik Group, 2025. Losgröße [WWW Document]. ProLogistik Group. URL <https://www.prologistik.com/logistik-lexikon/losgroesse/> (accessed 7.13.25).
- Proving Ground, 2022. Tracer Documentation [WWW Document]. Proving Ground Apps. URL <https://apps.provingground.io/docs/tracer-v1-0-documentation/> (accessed 8.5.25).
- Proving Ground, 2020. Tracer Plugin for Revit and IFC | Analyze BIM data in Power BI [WWW Document]. Proving Ground Apps. URL <https://apps.provingground.io/tracer/> (accessed 7.23.25).
- Revell, M., 2019. Low-code Development for IT Leaders [WWW Document]. URL <https://www.outsystems.com/de/tech-hub/low-code/> (accessed 11.1.24).
- Richardson, C., Rymer, J.R., 2016. Vendor Landscape: The Fractured, Fertile Terrain Of Low-Code Application Platforms [WWW Document].
- Richardson, C., Rymer, J.R., 2014. New Development Platforms Emerge For Customer-Facing Applications [WWW Document].
- Rymer, J.R., Koplowitz, T., 2019. M-Brain Is A Leader Among Market And Competitive Intelligence Platform Providers [WWW Document]. URL <https://reprintwavesummary.forrester.com/#/assets/WES0089/reprint-wavesummary?lang=en> (accessed 10.10.24).
- Sahay, A., Di Ruscio, D., Iovino, L., Pierantonio, A., 2023. *Analyzing business process management capabilities of low-code development platforms*. *Softw. Pract. Exp.* 53, 1036–1060. <https://doi.org/10.1002/spe.3177>
- Sanchis, R., García-Perales, Ó., Fraile, F., Poler, R., 2019. *Low-Code as Enabler of Digital Transformation in Manufacturing Industry*. *Appl. Sci.* 10, 12. <https://doi.org/10.3390/app10010012>
- Schalko, F., 2025. *Expertengespräch: Anwendung Low-Code Technologie beim Totalunternehmer*. Merenschwand. Geführt von Leah Lüscher am 01.04.2025.
- Scherer, P., Pfrunder, Marcel, Fischer, Martin, 2021. *Beyond VDC: digitales planen, bauen und bewirtschaften in der elektrobranche*. EIT.swiss, Zürich.
- Schildknecht, L., 2023. *Informationssysteme entwerfen*, Fachhochschule Nordwestschweiz, in: ISE. Presented at the Informationssysteme entwerfen, Fachhochschule Nordwestschweiz, Muttenz.
- Schweizer, R., 2025. *Expertengespräch: Anwendung Low-Code Technologie beim Totalunternehmer*. Laufenburg. Geführt von Leah Lüscher am 01.04.2025.
- Schweizerische Eidgenossenschaft, 2015. *Baujournal*, Bundesamt für Bauten und Logistik.
- Shah, A., 2020. Emptying Offices Prompt Adoption of Low-Code to Build Work Apps [WWW Document]. Wall Str. J. URL <https://www.wsj.com/articles/emptying-offices-prompt-adoption-of-low-code-to-build-work-apps-11589535001> (accessed 10.17.24).
- ShapeDiver, 2025. ShapeDiver | Cloud applications for Grasshopper [WWW Document]. ShapeDiver. URL <https://www.shapediver.com/> (accessed 8.5.25).
- SIA 102, 2020. *Ordnung für Leistungen und Honorar der Architektinnen und Architekten, Schweizerischer Ingenieur- und Architektenverein*.
- SIA 112, 2017. *Nachhaltiges Bauen im Hochbau, Schweizerischer Ingenieur- und Architektenverein*.
- SIA 118, 2013. *Allgemeine Bedingungen für Bauarbeiten, Schweizerischer Ingenieur- und Architektenverein*.
- SIA 400, 2000. *Planbearbeitung im Hochbau, Schweizerischer Ingenieur- und Architektenverein*.

- Siddiqui, F.H., Abdekhodae, A., Thaheem, M.J., 2023. *Taxonomy of Digital Skills Needed in the Construction Industry: A Literature Review*. Build. 2023 13. <https://doi.org/10.3390/buildings13112711>
- SIEMENS, 2024. BIM-Potenzial freisetzen: Wie AEC-Unternehmen multidisziplinäres Design für eine bessere Zusammenarbeit und eine schnellere Markteinführung nutzen können - Energie & Versorgungsunternehmen [WWW Document]. URL <https://blogs.sw.siemens.com/energy-utilities/2024/09/27/unleash-bim-potential-how-aec-companies-can-use-multidisciplinary-design-for-better-collaboration-faster-time-to-market/> (accessed 8.5.25).
- smapOne, 2025. smapOne - App-Baubericht [WWW Document]. URL <https://www.smapone.com/app-beispiele/> (accessed 6.12.25).
- SN EN ISO 16739-1, 2024. *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries - Part 1: Data schema (ISO 16739-1:2024)*, Schweizerische Normen-Vereinigung.
- Speckle, 2025. Speckle vs Autodesk Data Exchange [WWW Document]. URL <https://speckle.systems/blog/speckle-vs-autodesk-data-exchange/> (accessed 7.14.25).
- Spring, F., 2022. *Abstraktionsebenen der Dateiverwaltung*, in: ADV. Presented at the Automatisierte Datenverarbeitung, Muttenz.
- Stople, A., Steinsund, H., Iden, J., Bygstad, B., 2017. *LIGHTWEIGHT IT AND THE IT FUNCTION: EXPERIENCES FROM ROBOTIC PROCESS AUTOMATION IN A NORWEGIAN BANK*.
- Stucki, T., 2025. *Expertengespräch: Anwendung Low-Code Technologie bei der Bauleitung in einer Totalunternehmung*. Eschlikon. Geführt von Leah Lüscher am 14.04.2025.
- Terzic, N., 2025. *Expertengespräch: Anwendung Low-Code Technologie beim Totalunternehmer*. Dietikon. Geführt von Leah Lüscher am 11.04.2025.
- Tisi, M., Mottu, J.-M., Kolovos, D.S., de Lara, J., Guerra, E., Ruscio, D.D., Pierantonio, A., Wimmer, M., 2019. *Low-comote: Training the Next Generation of Experts in Scalable Low-Code Engineering Platforms*.
- Tracy van der Schyff, 2020. Power Platform: What are the connectors & triggers in Power Automate? [WWW Document]. URL <https://tracyvanderschyff.com/2020/03/05/what-are-the-connectors-triggers-in-power-automate/> (accessed 9.11.25).
- UiPath Inc, 2025. Sappi Streamlines Operations with Intelligent Automation | UiPath [WWW Document]. URL <https://www.uipath.com/resources/automation-case-studies/sappi-streamlines-operations-with-intelligent-automation> (accessed 8.17.25).
- Vesterlund, S., Falk, N.W., 2025. *Expertengespräch: Anwendung Low-Code Technologie mit Digitalen Spezialisten für das Bauwesen*. Kopenhagen. Geführt von Leah Lüscher am 24.04.2025.
- Vial, G., 2019. *Understanding digital transformation: A review and a research agenda*. J. Strateg. Inf. Syst., SI: Review issue 28, 118–144. <https://doi.org/10.1016/j.jsis.2019.01.003>
- Vikebø, E., Sydvold, L.B., 2019. An Inquiry into Low-Code Solutions in Institutions for Higher Education.
- Vincent, P., Iijima, K., Leow, A., Jain, A., Ray, S., Natis, Y., 2020. Magic Quadrant for Enterprise Low-Code Application Platforms [WWW Document]. Gartner. URL <https://www.gartner.com/en/documents/3991199> (accessed 10.13.24).
- Waszkowski, R., 2019. *Low-code platform for automating business processes in manufacturing*. IFAC-Pap., 13th IFAC Workshop on Intelligent Manufacturing Systems IMS 2019 52, 376–381. <https://doi.org/10.1016/j.ifacol.2019.10.060>
- Webster, J., Watson, R.T., 2002. *Analyzing the Past to Prepare for the Future: Writing a Literature Review*. MIS Q. 26, 2.
- Weidmann, N., Kirchhoff, J., Sauer, S., 2024. *Digitizing Processes in Manufacturing Companies via Low-Code Software*. Des. Equal. Justice 7–19. https://doi.org/10.1007/978-3-031-61688-4_2
- Wiendahl, H.-P., 2014. *Betriebsorganisation für Ingenieure*, 9., vollständig überarbeitete Auflage. ed. Hanser, München.

- Wong, J., Iijima, K., 2021. Gartner: Magic Quadrant for Enterprise Low-Code Application Platforms [WWW Document]. URL <https://www.gartner.com/en/documents/4005939>
- Woo, M., 2020. *The Rise of No/Low Code Software Development—No Experience Needed?* *Engineering* 6, 960–961. <https://doi.org/10.1016/j.eng.2020.07.007>

9 Abbildungs- und Tabellenverzeichnis

9.1 Abbildungen

Abbildung 1, schematische Struktur der Master-Thesis (eigene Abbildung)	3
Abbildung 2, Softwarelandschaft und ihre Zuordnung (eigene Abbildung)	4
Abbildung 3; Magic Quadrant for Enterprise Low-Code Application Platforms (Matvitsky, 2023)	6
Abbildung 4; Beispiel Blockly-Programmierung (Google, 2025)	8
Abbildung 5; Kernkomponenten Microsoft Power Platform (Microsoft, 2025a)	10
Abbildung 6; Power Automate Connectors (Tracy van der Schyff, 2020)	11
Abbildung 7; Tracer in der Umgebung von PowerBI (Proving Ground, 2020)	12
Abbildung 8; die verschiedenen Beteiligten (eigene Abbildung)	13
Abbildung 9; Die Unternehmung (eigene Darstellung)	14
Abbildung 10; Die Entwickelnden (eigene Darstellung)	14
Abbildung 11; Die Nutzenden (eigene Darstellung)	15
Abbildung 12; Beispiel einer Anwendung des Axialrasters (SIA 400, p. 18)	16
Abbildung 13; Auszug aus einer Projektplanung in ACC (eigene Abbildung)	16
Abbildung 14; Last Planner System (adaptiert von Scherer et al., 2021, pp. 80–81)	17
Abbildung 15; Bildschirmausschnitte der Low-Code-Applikation zur Dokumentation vom Baufortschritt (eigene Collage auf der Basis von smapOne Version 10.2, 08.11.2024)	20
Abbildung 16; Ansicht zum Editieren eines Bauprotokolls (Messerli Informatik, 2021)	20
Abbildung 17; Achsen im digitalen Bauwerksmodell (BuildingSmart, 2024)	21
Abbildung 18; Bauprozessplanung im IFC-Datenschema abgebildet als UML-Diagramm (adaptiert von Schildknecht, 2023)	22
Abbildung 19; Aufgabenabfolge (Building Smart, 2024)	23
Abbildung 20; Hauptgründe, um Low-Code development Plattformen zu nutzen (Sanchis et al., 2019, p. 3)	27
Abbildung 21; Gründe weshalb Low-Code nicht verwendet wird. (Sanchis et al., 2019, p. 3)	29
Abbildung 22, Ablauf zur Vorgehensweise (eigene Abbildung)	34
Abbildung 23; Komponenten zur mehrdimensionalen Betrachtung des Baufortschritts (eigene Abbildung)	38
Abbildung 24; Vorgehensprozess zur Integration eines IFC in einer Microsoft Power App (eigene Abbildung) ..	38
Abbildung 25; Vorgehensprozess zur Baufortschrittskontrolle (heute)	40
Abbildung 26; Angepasster Vorgehensprozess zur Baufortschrittskontrolle mit Einbezug eines digitalen Bauwerksmodells (morgen, neuer Prozess)	41
Abbildung 27; eigene Visualisierung vom «Welcome-Bildschirm» in Microsoft PowerApps	42
Abbildung 28; eigene Visualisierung von «ausgeführte Arbeiten: Übersicht» Variante 1 in Microsoft PowerApps	43
Abbildung 29; eigene Visualisierung von «Status bearbeiten» in Microsoft PowerApps	43
Abbildung 30; Visualisierung von Seite 1 «Unternehmerangaben» in Microsoft PowerApps	44
Abbildung 31; Visualisierung von Seite 2 «Unternehmerangaben» in Microsoft PowerApps	44

Abbildung 32; eigene Visualisierung von «ausgeführte Arbeiten: Übersicht» Variante 2 verknüpft mit dem Power BI Viewer via Tracer in Microsoft PowerApps	45
Abbildung 33; Benutzerführung in der Applikation (eigene Abbildung)	45
Abbildung 34; Differenzberechnung (Delta) zwischen geplanter zu tatsächlicher Bau-Dauer (eigene Abbildung)	46
Abbildung 35; Datenfluss in der Applikation (eigene Abbildung).....	47
Abbildung 36; Auswertung Nutzwertanalyse (eigene Abbildung).....	53
Abbildung 37; Verwendete Low-Code-Plattformen in der Baubranche auf der Basis von Expertenaussagen (eigene Abbildung)	58
Abbildung 38; Venn-Diagramm der Arbeitsgattungen (eigene Darstellung)	59
Abbildung 39; Auszug den unternehmenseigenen Apps aus APS (eigene Abbildung)	62
Abbildung 40; Auszug der unternehmenseigenen Applikation «Baufortschritts Simulator» aus APS (eigene Abbildung)	63

9.2 Tabellen

Tabelle 1; Vergleich verschiedene Low-Code-Plattformen.....	7
Tabelle 2; Vergleich verschiedene Low-Code-Plattformen mit IFC	9
Tabelle 3; versierte IT-Fachkräfte und ungeübte IT-Fachkräfte (Stople et al., 2017, p. 5).....	14
Tabelle 4; Vordefinierte Attribute von IfcTask für das Bauprogramm	24
Tabelle 5; Teilnehmende der Fokusgruppe, teilanonymisiert über eine generische ID	51
Tabelle 6, Verzeichnis verwendete Hilfsmittel	80

10 Anhang

- Anhang A:** (in separatem ZIP-Dateiordner)
- A0** - A0_MTH-Rechercheprotokoll.xlsx
 - A1** - Kap.5.1.3 - A1_MTH-ARC_Basis_01.ifc
 - A2** - Kap.5.1.3 - A2_IFC in CSV_var 6.txt -> 8-ung zu (*.py) ändern
 - A3** - Kap.5.1.3 - A3_MTH-Raumdaten_3.xlsx
 - A4** - Kap.5.1.3 - A4_AddTask_from-excel_to-IFC_4. txt -> 8-ung zu (*.py) ändern
 - A5** - Kap.5.1.3 - A5_MTH-Video Prototyp.mp4
 - A6** - Kap.5.1.3 - A6_MTH-App_Bilder.zip
 - A7** - Kap.5.1.3 - A7_MTH-ARC_tracer.db
 - A8** - Kap.5.3 - A8_MTH-Messgrößen.xlsx
 - A9** - Kap.5.1.4 - A9_MTH-Nutzwertanalyse.xlsx
 - A10** - Kap.5.1.4 - A10_MTH-Mentimeter Fragenkatalog.pdf
 - A11** - Kap.4.2 - A11_MTH -Interviewleitfaden_Bau-Experten.pdf
- Anhang B:** Verzeichnis verwendete Hilfsmittel
- Anhang C:** Vorlage Baujournal
- Anhang D:** Extraktion von Daten aus dem IFC in **ein CSV** (Python-Code)
- Anhang E:** Speicherung der Daten von der Datenbank in die IFC-Datei (Python-C)
- Anhang F:** Abstract English
- Anhang G:** Interview Totalunternehmer Schweiz
- Anhang H:** Interview beim Gleisbau-Unternehmen, Schweiz
- Anhang I:** Interview mit Totalunternehmer, Schweiz
- Anhang J:** Interview mit Totalunternehmer, Schweiz
- Anhang K:** Interview bei Bauleitung in einer Totalunternehmung, Schweiz
- Anhang L:** Interview beim Totalunternehmer, Schweiz
- Anhang M:** Interview beim Elektrotechnik-Unternehmen, Schweiz
- Anhang N:** Interview mit Digitalen Spezialisten für das Bauwesen, Dänemark
- Anhang O:** Interview mit einem Digital Consulting-Unternehmen, Schweiz
- Anhang P:** Link Prototyp

Anhang B Verzeichnis verwendete Hilfsmittel

Vom Verfasser genutzte Anwendungen, Softwares und Hilfsmittel zur Erstellung der Master-Thesis.

Tabelle 6, Verzeichnis verwendete Hilfsmittel

Hilfsmittel	Teile / Stelle (n) in der Arbeit	Art des Einsatzes
Swisscovery plus FHNW	Recherche gemäss Rechercheprotokoll	Suchmaschine für akademische Quellen: (umfassendes Rechercheprotokoll ist dem Anhang zu entnehmen, siehe 'A0 Rechercheprotokoll_MTH_lclu.xlsx')
SNV-Connect	Alle aktuellen Normen	Zugriff und Download aller verwendeten Normen
Google-Suchmaschine	Recherche gemäss Rechercheprotokoll	Recherche von technischen Grundlagen, graue Literatur (weitere Ausführungen sind dem Anhang zu entnehmen, siehe 'A0 Rechercheprotokoll_MTH_lclu.xlsx')
DeepL Translate	Quellen, Papers in Englischer Sprache	Übersetzung von Englisch auf Deutsch bei Fach-Publikation
DeepL Write	Gesamte Arbeit	Umformulieren und Kontrolle Rechtschreibung
ChatGPT 40	Kapitel 5.1.3	Unterstützung bei Struktur und Konzeptentwicklung für den Python-Code
IfcOpenshell, IfcOpenshell.geom v.0.8.2	Kapitel 5.1.3	Konvertierung der IFC-Datei in ein CSV und Konvertierung von Excel in IFC
pandas v.2.3.1	Kapitel 5.1.3	Konvertierung der IFC-Datei in ein CSV und Konvertierung von Excel in IFC
Openpyxl v.3.1.5	Kapitel 5.1.3	Konvertierung der IFC-Datei in ein CSV und Konvertierung von Excel in IFC
Python v.3.10.7150.0	Kapitel 5.1.3	Erstellung Code in traditioneller Codesprache Python.
Microsoft (MS)-Word v.2507 Build 16.0.19029.20136, 64 Bit	Gesamte Arbeit	Erstellung und Formatierung der finalen Dokumentation
Visual Studio Code v.1.103.2	Kapitel 5.1.3	Umsetzung Code zur Umsetzung vom Prototyp
MS-Editor, Inside Word	Gesamte Arbeit	Ergebnis vor der Prüfung: 68% Ergebnis nach der Prüfung: 92% Dabei wurden folgende Korrekturen über-/vorgenommen: Rechtschreibung:574 Stk. Grammatik: 107 Stk. Formelle Sprache:157 Stk. Interpunktion, 4 Stk. Prägnanz: 32 Stk. Wortwahl: 8 Stk.
MS-Excel v.2507 Build 16.0.19029.20136, 64 Bit	Dokumente	Erstellen von Auswertungen & Beurteilungsraster
Zotero v.7.0.24	Gesamte Arbeit	Literaturverwaltungsprogramm zum Sammeln, Verwalten und Zitieren aller Quellen

Hilfsmittel	Teile / Stelle (n) in der Arbeit	Art des Einsatzes
Microsoft Power Apps v.3.25082.8.0	Kapitel 5.1.3.2	Erstellung App
Microsoft Power BI V.2.146.1026.0	Kapitel 5.1.3.2	Erstellung Dashboard
MS-PowerPoint v.2508 Build 16.0.19127.20192, 64 Bit	Alle Grafiken	Design & Entwurf aller eigens erstellten Grafiken, inkl. der darin frei enthaltenen Piktogramme
MS-Teams v.25198.1112.3855.2900	Meetings, Fokusgruppen	Durchführung von online Besprechun- gen, Diskussionen & Ideenaustausch
Draw.io (Browser)	Alle Prozessdiagramme	Design & Entwurf aller eigens erstellten Prozessdiagramme
Archicad 27	Kapitel 5.1.3.2 MTH-ARC_Basis_01.ifc	Modellierung vom Grundmodell, mit anschliessendem Export der notwendi- gen Dateien für die Weiterverarbeitung im Prototyp

Anhang C Vorlage Baujournal

Muster 1 zu Anweisungen für beauftragte Bauleitungen

(Die Muster dienen als formale Vorlagen und nicht als grafische Vorgaben)

MUSTER ARCHITEKTEN GMBH

Musterstrasse 15
1234 Musterhausen

Baujournal: Neubau Musterprojekt		
Datum:	<input type="checkbox"/> Mo. <input type="checkbox"/> Di <input type="checkbox"/> Mi <input type="checkbox"/> Do <input type="checkbox"/> Fr <input type="checkbox"/> Sa <input type="checkbox"/> So
Wetter:	<input type="checkbox"/> Schnee <input type="checkbox"/> Regen <input type="checkbox"/> Wolken <input type="checkbox"/> Sonne <input type="checkbox"/> Wechselhaft <input type="checkbox"/>	
Temperaturen min. ° C: ° C	Niederschlagsmenge mm/m2:
Temperaturen max. ° C: ° C	Luftfeuchte %:

A) Anwesende Unternehmer:	Arbeitsgattung:	Arbeitskräfte:
1		
2		
3		
4		
5		
6		
7		
8		

B) Ausgeführte Tätigkeiten der Unternehmer:	Bereich:
1	
2	
3	
4	
5	
6	
7	
8	

C) Begehung der Baustelle durch Dritte:	
---	--

D) Aussergewöhnliche Vorkommnisse, Unfälle etc.:	
--	--

E) Anordnungen auf der Baustelle durch Bauleitung:	
--	--

F) Anordnungen auf der Baustelle durch Architekten:	
---	--

G) Planlieferung an Unternehmer:	
----------------------------------	--

Ort / Datum: Visum Bauleitung:

Anhang D Extraktion von Daten aus dem IFC in ein CSV (Python-Code)

Mich interessierte, ob ich mit Grundlagenwissen über Programmierung in Unterstützung mit ChatGPT ein IFC als ein Excel speichern kann. Dies im Zusammenhang mit den „Citizen Developer“ welche auch nur wenig wissen über Programmieren aufweisen und eine Applikation bauen möchten.

Plattform: www.chatgpt.com

Anfragezeitpunkt: 10.07.2025, 08:58

Gesendete Nachricht an ChatGPT-4o:

Erstelle ein Python-Skript, das aus einer IFC-Datei (z. B. MTH-ARC.ifc) ausgewählte Bauteile (IfcSpace, IfcWall, IfcWallStandardCase) ausliest und ihre Kerndaten in eine CSV exportiert. Die CSV enthält neben Geometriedaten auch leere/optionale Spalten für spätere Termin- und Taskfelder. Im Code werden keine Umlaute verwendet (z. B. Laenge, Hoehe, Dicke).

Grundstruktur für den Ablauf vom Programm:

- 0) Module importieren
- 1) Dateipfade für die Arbeitsdateien
- 2) IFC-Datei oeffnen
- 3) Daten aus Dokument lesen
- 4) Datensätze durchlaufen
 - 4.1) Globalld für Bauteilidentifizierung lesen
 - 4.2) Bauteilname
 - 4.2) Geschoss
 - 4.3) Länge, Höhe und Dicke
 - 4.4) Nächsten Datensatz in Datentabelle lesen
- 5) Ressourcen freigeben
- 6) CSV-Datei mit neuen Namen speichern

Nutze dabei die nachfolgenden, zusätzlichen Ergänzungen als Unterstützung:

```
#0) Module importieren
#
#1) Dateipfade für die Arbeitsdateien
IFC_PATH = Path(r"C:\Users\Lel\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\Grundlagen
App\BSc_Architekturmodell.ifc")
CSV_PATH = Path(r"C:\Users\Lel\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\umgeschriebenes
IFC\ifc_export.csv")
#
#2) Hilfsfunktionen
get_container(element, target_type="IfcBuildingStorey"): Ermittelt das zugehoerige Geschoss.
get_psets(element): Liest PropertySets und extrahiert verfuegbare Eigenschaften (inkl. BaseQuantities).
```

get_bbox_dims(element, settings): Berechnet ersatzweise die Bounding-Box-Abmessungen (x, y, z), falls keine quantitativen Angaben vorhanden sind.

#

#3) Exportfunktion export_ifc(ifc_path, csv_path)

IFC-Modell oeffnen und Geometrie-Settings setzen (Weltkoordinaten).

Elemente sammeln: IfcSpace, IfcWall, IfcWallStandardCase.

Pro Element:

PropertySets lesen, bevorzugt BaseQuantities (Length/Height/Width bzw. Thickness).

Falls fehlt: Dimensionen aus der Bounding-Box herleiten.

Container (Geschoss) und Raumnamen bestimmen.

Datensatz zusammenstellen mit:

GlobalId, Bauteil, Geschoss, Raum, Laenge, Hoehe, Dicke plus Termin-/Task-Spalten (zunaechst leer):

DurationType, ScheduleDuration, ScheduleStart, ScheduleFinish,

EarlyStart, EarlyFinish, LateStart, LateFinish, FreeFloat,

TotalFloat, IsCritical, StatusTime, ActualDuration,

ActualStart, ActualFinish, RemainingTime, Completion.

Alle Datensaeetze als CSV nach CSV_PATH schreiben (Semikolon-Trennung).

#

#4) CSV-Datei speichern

export_ifc(IFC_PATH, CSV_PATH) ausfuehren und erfolgreichen Export melden.

Erhaltene Antwort von ChatGPT-4o:

```
#!/usr/bin/env python3
...
IFC → CSV-Exporter (IfcSpace & IfcWall)

Extrahiert GUID, Geschoss, Bezeichnung, Länge, Höhe und Dicke
aus IfcSpace- und IfcWall-Objekten einer IFC-Datei
und speichert alles semikolon-getrennt als CSV.

Benötigte Pakete:
  pip install ifcopenshell pandas
...

from pathlib import Path
import ifcopenshell
import pandas as pd

# -----
# 1) DATEIPFADE – anpassen bei Bedarf
# -----
IFC_PATH = Path(r"C:\Users\Lele\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\Grundlagen App\BSc_Architekturmodell.ifc")
CSV_PATH = Path(r"C:\Users\Lele\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\umgeschriebenes IFC\ifc_export.csv")

# -----
# 2) HILFSFUNKTIONEN
# -----
def get_container(element, target_type="IfcBuildingStorey"):
    """Liefert das erste übergeordnete Element des gewünschten Typs (z. B. Geschoss)."""
    if not hasattr(element, "ContainedInStructure"):
        return None
    for rel in element.ContainedInStructure:
        container = rel.RelatingStructure
```

```

if container and container.is_a(target_type):
    return container
return None

def get_psets(element, include_inherited=True):
    """Liest alle PropertySets (inkl. vererbter) als geschachteltes Dict aus."""
    psets = {}
    if hasattr(element, "IsDefinedBy"):
        for rel in element.IsDefinedBy:
            if not rel.is_a("IfcRelDefinesByProperties"):
                continue
            prop_def = rel.RelatingPropertyDefinition
            if prop_def and prop_def.is_a("IfcPropertySet"):
                props = {
                    prop.Name: getattr(prop.NominalValue, "WrappedValue", None)
                    for prop in prop_def.HasProperties
                    if hasattr(prop, "NominalValue") and prop.NominalValue
                }
                psets[prop_def.Name] = props
    if include_inherited and hasattr(element, "Decomposes"):
        for rel in element.Decomposes:
            parent = rel.RelatingObject
            if parent:
                psets.update(get_psets(parent, include_inherited=True))
    return psets

def extract_geom_fallback(element):
    """Gibt BoundingBox-Maße zurück, falls BaseQuantities fehlen."""
    try:
        bb = element.BoundingBox()
        return round(bb.XDim, 3), round(bb.ZDim, 3), round(bb.YDim, 3)
    except Exception:
        return None, None, None

# -----
# 3) DATEN EXTRAHIEREN UND EXPORTIEREN
# -----

def export_ifc(ifc_path: Path, csv_path: Path) -> None:
    print(f"📂 Lade IFC-Datei: {ifc_path}")
    model = ifcopenshell.open(str(ifc_path))

    rows: list[dict] = []

    # ----- SPACES -----
    spaces = model.by_type("IfcSpace")
    print(f"📁 Verarbeite {len(spaces)} IfcSpace-Objekte ...")
    for space in spaces:
        psets = get_psets(space)
        bq = psets.get("BaseQuantities", {})
        bb_length, bb_height, bb_thick = extract_geom_fallback(space)
        length = bq.get("Length") or bb_length
        height = bq.get("Height") or bb_height

        container = get_container(space)
        geschoss_name = container.Name if container and hasattr(container, "Name") else ""

        rows.append(dict(
            Typ="Space",
            GUID=space.GlobalId,
            Geschoss=geschoss_name,
            Bezeichnung=space.LongName or space.Name or "",
            Länge=length,
            Höhe=height,
            Dicke=None,
        ))

    # ----- WALLS -----
    wall_types = ("IfcWall", "IfcWallStandardCase")
    walls = [w for t in wall_types for w in model.by_type(t)]
    print(f"📁 Verarbeite {len(walls)} IfcWall-Objekte ...")
    for wall in walls:
        psets = get_psets(wall)
        bq = psets.get("BaseQuantities", {})
        bb_length, bb_height, bb_thick = extract_geom_fallback(wall)

```

```

length = bq.get("Length") or bq.get("Width") or bb_length
height = bq.get("Height") or bb_height
thickness = (
    bq.get("Width")
    or bq.get("Thickness")
    or psets.get("Pset_WallCommon", {}).get("Thickness")
    or bb_thick
)

container = get_container(wall)
geschoss_name = container.Name if container and hasattr(container, "Name") else ""

rows.append(dict(
    Typ="Wall",
    GUID=wall.GlobalId,
    Geschoss=geschoss_name,
    Bezeichnung=wall.Name or "",
    Länge=length,
    Höhe=height,
    Dicke=thickness,
))

print(f"📊 Insgesamt {len(rows)} Datensätze gesammelt – schreibe CSV ...")
df = pd.DataFrame(rows)
df.to_csv(csv_path, index=False, sep=";")
print(f"✅ Export abgeschlossen → Datei gespeichert unter: {csv_path}")

# -----
# 4) STARTPUNKT
# -----
if __name__ == "__main__":
    export_ifc(IFC_PATH, CSV_PATH)

```

Beurteilung der erhaltenen Antwort von ChatGPT-4o durch den Verfasser:

Code ist funktionsfähig gibt anfänglich aber Fehler aus. Mit etwas Geduld und ein paar Iterationen lässt sich dies aber bewerkstelligen.

Anhang E Speicherung der Daten von der Datenbank in die IFC-Datei (Python-Code)

Um den Prozess ganzheitlich abbilden zu können, müssen die Informationen des Excel-Datei auch wieder in ein IFC gespeichert werden können.

Plattform: www.chatgpt.com

Anfragezeitpunkt: 29.05.2025, 15:43

Gesendete Nachricht an ChatGPT-4o:

Erstelle mir einen Python-Code mit der Funktionalität "Hinzufügen von Taskzuständen ein eine bestehende IFC-Datei auf der Basis von einer Excel-Datei" auf der Basis vom nachfolgenden Input, vermeide im Code sämtliche Umlaute:

Grundstruktur für den Ablauf vom Programm:

0) Module importieren

1) Dateipfade für die Arbeitsdateien

2) IFC-Datei und Datenbank (Excel) oeffnen

3) Daten aus Datenbank lesen, um diese dann in IFC-Datei zu übertragen

4) Datensätze durchlaufen und Task-Zustaende in zugehörigen IFC-Datensatz schreiben

4.1) Globalld für Bauteilidentifizierung lesen

4.2) Task-Instanz anhand der Globalld lesen

4.2.n) bei nichtvorhandensein einer Task-Instanz, wird diese auf der Basis der "IFC4X3_ADD2"-Dokumentation (https://standards.buildingsmart.org/IFC/RELEASE/IFC4_3/) hinzugefügt

4.3) Task der Globalld-Instanz neu schreiben

4.4) Nächsten Datensatz in Datentabelle lesen

5) Ressourcen freigeben

6) IFC-Datei mit neuen Namen speichern

Nutze dabei die nachfolgenden, zusätzlichen Ergänzungen als Unterstützung:

```
# 0) Module importieren
```

```
#
```

```
# 1) Dateipfade für die Arbeitsdateien
```

```
input_excel_filepath = r"C:\Users\Lel\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\umgeschriebenes IFC\MTH-Raumdaten_3.xlsx"
```

```
# C:\Users\Lel\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\umgeschriebenes IFC\MTH-ARC_spaces.ifc.xlsx"
```

```
ifc_filepath = r"C:\Users\Lel\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\Grundlagen App\MTH-ARC_.ifc" #IfcTasks_Export - Kopie.ifc" #MTH-ARC_.ifc"
```

```
# C:\Users\Lel\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\Grundlagen App\MTH-ARC_.ifc"
```

```
output_ifc = r"C:\Users\Lel\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\Grundlagen App\IfcTasks_withtime_Export.ifc"
```

```

#C:\Users\Lele\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\umgeschriebenes IFC\IfcTasks_Export.ifc"#
#
# 2) IFC-Datei und Datenbank (Excel) oeffnen
ifcfile = ifcopenshell.open(ifcfilepath)
dbconn = sqlite3.connect(dbpath)
dbcursor = dbconn.cursor()
#
# 3) Daten aus Datenbank lesen, um diese dann in IFC-Datei zu übertragen
#
# 4) Datensätze durchlaufen und Task-Zustände in zugehörigen IFC-Datensatz schreiben
# 4.1) Globalld für Bauteilidentifizierung lesen
globalid = raum[0]
# 4.2) Task-Instanz anhand der Globalld lesen
ifcspace = ifcfile.by_guid(globalid)
# 4.3) Task der Globalld-Instanz neu schreiben
ifcspace.Name = raum[1]
# 4.4) Nächsten Datensatz in Datentabelle lesen
raum = dbcursor.fetchone()
#
# 5) Ressourcen freigeben
dbcursor.close()
dbconn.close()
#
# 6) IFC-Datei mit neuen Namen speichern
ifcfile.write(output_ifc)

```

Erhaltene Antwort von ChatGPT-4o:

Hier ist ein vollständiger Python-Code, der deine Vorgaben Schritt für Schritt umsetzt. Er nutzt ifcopenshell zum Bearbeiten der IFC-Datei und openpyxl zum Auslesen der Excel-Datei. Er vermeidet Umlaute und erfüllt deine beschriebene Programmlogik.

```

# 0) Module importieren
import ifcopenshell
import openpyxl
import uuid
from datetime import datetime

# 1) Dateipfade fuer die Arbeitsdateien
input_excel_filepath = r"C:\Users\Lele\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\umgeschriebenes IFC\MTH-Raumdaten_3.xlsx"
# C:\Users\Lele\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\umgeschriebenes IFC\MTH-ARC_spaces.ifc.xlsx"
ifc_filepath = r"C:\Users\Lele\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\Grundlagen App\MTH-ARC_ifc" #IfcTasks_Export - Kopie.ifc"
#MTH-ARC_ifc"
# C:\Users\Lele\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\Grundlagen App\MTH-ARC_ifc"
output_ifc = r"C:\Users\Lele\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\Grundlagen App\IfcTasks_withtime_Export.ifc"
#C:\Users\Lele\OneDrive\Dokumente\MTH-Masterthesis\06 Produkt\umgeschriebenes IFC\IfcTasks_Export.ifc"
# 2) IFC-Datei und Excel-Daten oeffnen
ifcfile = ifcopenshell.open(ifc_filepath)
wb = openpyxl.load_workbook(input_excel_filepath)
sheet = wb.active

```

```

# 3) Daten aus Excel lesen
rows = list(sheet.iter_rows(min_row=2, values_only=True)) # Annahme: Erste Zeile = Kopfzeile

# 4) Datensätze durchlaufen und Task-Zustände einfügen oder aktualisieren
for row in rows:
    globalid = row[0]
    task_name = row[1]
    task_status = row[2] if len(row) > 2 else "UNDEFINED"

    if not globalid or not task_name:
        continue

    # 4.2) IFC-Instanz anhand GlobalId finden
    related_object = ifcfile.by_guid(globalid)

    if not related_object:
        print(f"Element mit GlobalId {globalid} nicht gefunden.")
        continue

    # 4.2.n) Neue Task-Instanz hinzufügen, falls nicht vorhanden
    # Es wird angenommen, dass keine Task-Instanzen vorhanden sind, also wird immer eine neue erstellt

    # IFC-Zeitformat und eindeutige Kennung erstellen
    creation_time = datetime.utcnow().isoformat()
    task_guid = ifcopenshell.guid.new()
    task_object = ifcfile.create_entity(
        "IfcTask",
        GlobalId=task_guid,
        Name=task_name,
        ObjectType="Task",
        Identification=f"TASK-{globalid}",
        Status=task_status,
        TaskTime=None, # Optional: ifcopenshell kann auch IfcTaskTime erzeugen
        PredefinedType="NOTDEFINED"
    )

    # Beziehung erstellen zwischen Task und Space (oder anderem Element)
    rel_guid = ifcopenshell.guid.new()
    ifcfile.create_entity(
        "IfcRelAssignsToControl",
        GlobalId=rel_guid,
        RelatingControl=task_object,
        RelatedObjects=[related_object]
    )

# 5) Ressourcen freigeben
wb.close()

# 6) IFC-Datei mit neuen Namen speichern
ifcfile.write(output_ifc)

print("IFC-Datei erfolgreich aktualisiert und gespeichert.")

```

Beurteilung der erhaltenen Antwort von ChatGPT-4o durch den Verfasser:

Code ist funktionsfähig gibt anfänglich aber Fehler aus. Mit etwas Geduld und ein paar Iterationen lässt sich dies aber bewerkstelligen.

Anhang F Abstract English

The construction industry is undergoing a major digital transformation. Building Information Modelling enables data-based mapping of a building's life cycle, but users face limits due to complex applications. Low-code platforms gain attention as they allow application-specific software development without deep programming knowledge using simple code blocks.

This thesis examines how a digital building model and low-code platforms can be combined to increase efficiency and innovation in the construction industry. Based on a literature review and unstructured expert interviews, the opportunities and challenges of combined use are analysed using a prototype. This is followed by a discussion and utility analysis within the framework of a focus group to validate the prototype.

Expert interviews show that low-code is mainly used for internal process optimisation, while data governance and scalability are controversial. In planning, dashboards compare project data or costs. Apps for managing small databases on-site are popular and often implemented with Microsoft Power Apps.

The product is an application that allows construction progress to be edited on the go, component by component, as part of a construction log. This tool allows adjustments to be made to the planning in an agile manner and implemented with shorter intervals between the comparison of actual and target values. The standardised IFC data schema makes the data accessible to all parties involved. This closes the interface between construction management and planning, ensuring greater security and improved collaboration. Likewise, the consideration of normative principles creates a scalable application.

Low-code platforms, IFC, digital building model, construction progress, construction log, Microsoft Power Apps

Anhang G Interview Totalunternehmer Schweiz

Stichwortprotokoll (Pfister, 2024)

Frage	Aussage
Welchen Anwendungsfall habt ihr bereits umgesetzt?	Wir haben eine Applikation zu den Schichten im Tunnel mit den verbauten Materialien zu einzelnen Abschnitten erstellt.
Welche Weiteren Use-Cases habt ihr bereits im Einsatz?	Wir haben eine App zur Messung von Wertschöpfenden/nicht Wertschöpfenden erstellt. Damit kann erkannt werden welche verschwendeten Tätigkeiten täglich gemacht werden. Diese ist bereits im Einsatz.
Was ist deiner Meinung wichtig, im Zusammenhang mit Low-Code, damit die Art des Bauens verändert werden kann?	Die Technologie allein unterstützt die Digitalisierung nicht. Es bedarf der Anpassung des ganzen Prozesses.
Wie geht ihr mit Data Gouvernment um?	die Data Gouvernment und die damit verbundenen Berechtigungen der Benutzer hat entscheidenden Einfluss auf die Daten und ihre Lebensdauer.
Welches Tool würde euch auf der Baustelle unterstützen?	Fotos zu einzelnen Abschnitten beispielsweise in einem Tunnel wäre eine grosse Unterstützung auf der Baustelle.
Wie viele Leute beschäftigen sich bei Implenia mit Low-Code?	Es sind zwei Leute aus der IT, welche sich ausschliesslich mit Low-Code befassen (Seit 2020) Es gibt zusätzlich ein erweitertes Team aus der Planung welche sich damit auseinandersetzen. Dabei sind die Erstellten gleichzeitig auch die Nutzer.
Wer ist es, welche mit Use-Cases, nach Applikationen fragen?	Bauführer erstellen vereinzelt auch schon Low-Code-Anwendungen. Diese sind in der Vorbildung meist Elektro-Ingenieur oder ähnliches. Dies ist aber eher die Ausnahme.
Welche Low-Code-Plattformen setzt ihr in der Unternehmung ein?	Hauptsächlich Microsoft Power Platform.
Was sind Anwendungen mit Microsoft Power Platform?	Beispielsweise sind das Projekt-Anträge, welche über Power BI ausgelesen werden und es so ermöglichen gleichmässige Vergleiche zu ziehen. Dieses Dashboard ist bereits im Einsatz.
Was ist eines der schwierigsten Themen in Bezug auf Low-Code?	Am schwierigsten ist sicherlich die Durchgängigkeit der Daten.
Wann ist eine Umsetzung mit Low-Code schwierig?	Low-Code-Anwendungen spezifisch für Branchen-Anwendungen sind schwierig zu standardisieren trotzdem aber sehr wichtig für die Unternehmung. Denn nur so können sie ihr Potenzial zeigen.
Wann ist eine Umsetzung mit Low-Code sinnvoll?	Allgemeine Anwendungen wie kleine Automatisierungen sind sicher gut möglich. Ebenfalls können gut auch Firma-übergreifende Anwendungen sinnvoll sein, welche danach weiterverwendet werden.
Welchen Anwendungsfall habt ihr bereits umgesetzt?	Ebenfalls haben wir eine App zur Kontrolle des Lagerbestands. Damit können wir die vorhandenen Bauteile einsehen und bearbeiten. Dass ist sinnvoll da damit nicht jede Woche ein Mitarbeiter, während 2h Anker zählen muss. Es ist vergleichbar mit einer Art internes Ebay.
Wann wird Low-Code verwendet und nicht Hard-Code?	Meist werden Applikation in Low-Code erstellt, um in kurzer Zeit den Nutzen zu testen. Manchmal werden sie dann der IT übergeben, welche daraus ein traditioneller Code entwickelt. Das kommt aber selten vor.
Was kann Low-Code verbessern im Vergleich zur heutigen Situation?	Wir erkennen häufig in der Baubranche eine starke Fragmentierung der Projekte, denn immer wieder Informationen verloren gehen. Dies hat Verschiedene Gründe, weshalb Projekte Informationsverluste haben: Dies können einerseits andere Personen

	sein, oder weil es ein anderes Projekt ist. Wir hätten meist die Technik und das Wissen dazu, allerdings sind wir meist abhängig von Auftraggeber und was dieser Bestellt.
Wie werden die Mitarbeiter für die Verwendung eines Tools unterstützt?	Neue Tools werden in einer Schulung den Mitarbeitern gezeigt, was sehr wichtig ist, um die gewünschte Datenqualität zu erhalten. Nur wenn alle Mitarbeiter die Tools gleich nutzen und das gleiche Verständnis haben, können Vergleiche angestellt werden.
Welche Herausforderungen könnten durch die Anwendung dieser Technologie auftreten?	Branchenspezifische Low-Code-Anwendungen stossen dagegen regelmässig an Grenzen, weil sie aufwändige Anpassungen an unternehmensinterne Standards, Schnittstellen und Compliance-Richtlinien erfordern.
Wie sollte eine ideale App für den Baustellenalltag gestaltet sein, um die Arbeit effizienter zu gestalten?	Die ideale App ist wie Instagram. Sie soll wie Low-Code nicht schwierig sein und einfach anzuwenden. Somit soll sie selbsterklärend sein und den Nutzern ohne Schulung übergeben werden können.
Wo sehen Sie die grössten Potenziale, um mit Low-Code im Bauwesen Prozesse zu automatisieren und zu optimieren?	Mit Low-Code als Grundlage können Veränderung durch Nutzer möglich gemacht werden, weil der Ersteller sehr nahe an Endnutzer ist und Needs in kurzen Wegen entstehen können.

Anhang H Interview beim Gleisbau-Unternehmen, Schweiz

Stichwortprotokoll (Bosshart, 2025)

Frage	Aussage
Wo sind konkrete Anwendungsfälle?	Wir haben eine Applikation erstellt bei welcher ein Leistungsausmass gemacht, beispielsweise ein Arbeitsprogramm erstellt wird. Darin wird die Gleisbaumaschine erfasst, deren Auswertungen werden uns per E-Mail geschickt in das Postfach und dann als XML-File automatisiert, heruntergeladen und anschliessend in der Datenbank gespeichert.
Wie setzt sich euer Team zusammen?	Ich gehöre zu einem Entwicklungsteam nahe mit der Fahrbahn verbunden. Dieses Entwicklungsteam beschäftigt sich ausschliesslich mit IT-Lösungen und ist auch der IT angegliedert.
Wer kommt mit Use Cases?	Viele haben sehr Berührungängste: ein paar jüngere, die Nachkommen und schon Erfahrungen gemacht haben, sei es im Studium mit Low Code und denken, sie probieren einfach mal, ob sie das können und dann vielleicht eine Automatisierung für sich erstellen und sich dann so der Arbeitsablauf ein wenig vereinfachen, oder? Es dauert bis man genug Leute haben, die davon überzeugt sind, dass es dann sich verbreitet oder aber die ersten paar Early Adapters zu finden.
Was sind Herausforderungen?	Jeder macht eher Automatisierungen für sich selbst, aber nicht im Konzern geteilt.
Was habt ihr mit Low-Code erstellt?	Wir haben eine Applikation erstellt, welche eine Art ein Dashboard ist, wo wichtige Informationen zusammenfliessen. Beispielsweise Kennzahlen.
Was habt ihr mit Low-Code erstellt?	Ein Anwendungsbeispiel ist eine App, mit welcher viele verschiedene Varianten von Bauabläufen bezüglich Kosten gegenübergestellt werden können. Die Applikation bezieht sich auf Georeferenzierung in Gleiskilometer. Sie ist aber nicht mit einem Modell verbunden.
Könnten Applikationen für die gesamte Firma als IT-Team erstellt werden?	Ich glaube eben, das schaffst du nicht, wenn du weiter weg bist, von der Ausführung. Du musst in die Linie der Ausführung

	kommen. Sonst wirst du keine guten Automatisierungen erstellen können, weil das gemeine Vokabular fehlt.
Was für eine Anwendung würdest du dir wünschen?	Ein guter Ansatz wäre es, wenn die Low-Code-Applikation mit den IFC-Dateien kompatibel wäre und am Abend im Büro gemütlich der Baufortschritt abgefragt werden könnte.
Warum verwendet ihr Low-Code und nicht Hardcodiert?	Wir sind ein Team, welches eine Art Schnell-Entwickler besteht und eine Erweiterung der IT sind. Wir haben das Ziel, dass wenn Ideen auf der Baustelle entstehen, welche man automatisieren könnte, dass wir diese dann ausführen. Dieser Ansatz ist für uns sinnvoll, weil es natürlich viele Verständnisprobleme gibt, wenn Ausführende mit den IT-Spezialisten reden. Beide reden die gleiche Sprache, trotzdem fordert es ein sehr langer Prozess mit nur wenig Ergebnissen.
Warum wird die Low-Code-Technologie nicht mehr verwendet?	Baubranche ist überall ein wenig langsamer. Darum dauert Digitalisierung so lange. Deshalb erstellen wir mit Low-Code Prototypen, welche schnell getestet werden können.
Was sind technische Herausforderungen für den Einsatz von Low-Code	Der ganze Prozess muss rundum standardisiert werden. Dies beginnt bei der Anforderungsanalyse, Modellierung, Bereitstellung und Wartung.
Wie steht ihr zum Thema Standardisierung?	Es ist wichtig, dass wir es ein Format bringen, welches lesbar ist. Sonst hast du einen Projektleiter ist, der ist so im Detail, dass du niemals über die gleichen Dinge sprechen kannst.
Was sind weitere Use-Cases?	Wir haben eine Automatisierung erstellt, mit welcher ich auf eine SQL-Datenbank zugreifen kann, welche mir die Informationen automatisch in ein Dashboard lädt.
Verwendet ihr die Power Apps von Microsoft und für was?	Im Corporate Bereich brauchen wir die Power Apps. Also beispielsweise wo man zum Beispiel gewisse Sachen automatisiert wie Kleider bestellen oder ähnliches.
Wie geht ihr mit den Daten um, damit die Datendurchgängigkeit gewährleistet ist?	Also Datendurchgängigkeit ist ein grosses Thema bei uns. Wir haben einen einheitlichen Datenpool, welcher für unsere Enterprise Analytics funktioniert. Dort fliessen die wichtigsten Informationen zusammen. Die Datendurchgängigkeit wird ab 2027 mit neuem SAP durchgängig gelöst sein. Im Momentan fehlt noch das Stück, dass alles verbindet.
Wie viele Rollen gibt es bei euch, bei der Erfassung von Daten in den Applikationen?	Bei Applikationen gibt es nur eine Rolle. Dies ist so weil die IT-Spezialisten die Applikation erstellt und Bauleiter sie nutzen. Somit gibt es nur die Rolle des Nutzers und des Erstellers.
Welche Chancen bietet Low-Code?	Low-Code hat als Übergangslösung ganz klar die Chance viele Arbeitsschritte, welche wir häufig machen zu automatisieren. Dies kann uns helfen, bis alle Prozesse standardisiert sind.
Welche Chancen bietet Low-Code noch?	Mit Low-Code können Lösungen, schnell umgesetzt werden und können als Prototypen getestet werden.
Wieso seid ihr als IT-nahes Team dem Team Fahrbahn angegliedert und wieso gibt es das für andere Teams nicht?	Low-Code wird bei Team Fahrbahn eingesetzt, weil dort viel Geld eingespart werden kann. Deshalb gibt es ein eigenes Sub-Team der IT für die Fahrbahn.
Wie werden bei euch Use Cases erfasst?	Needs werden von Ausführenden erstellt und die IT setzt es um. Leute werden so direkt überzeugt und sind begeistert das Tool, um zu setzen. Bevor Applikation umgesetzt wird, wird mit den Leuten ein Proof of Concept erstellt.
Macht ihr Schulungen bei der Einführung von Applikationen?	Schulung des Personals ist wichtig für Erfolg. Dies ist zwar aufwändig, zahlt sich aber aus.

Anhang I Interview mit Totalunternehmer, Schweiz

Stichwortprotokoll (Schweizer, 2025)

Frage	Aussage
Arbeitet ihr mir Low-Code-Applikationen?	Ja, wir arbeiten mit Low-Code-Applikationen, hauptsächlich im Microsoft-Umfeld :
Was ist eine Chance bei der Verwendung von Low-Code?	Low-Code ist recht cool, weil du sehr gut über die Microsoft Inter-netdienste anbinden kannst und das ganz Ökosystem ermöglicht einem sehr viel, was sonst schwierig machbar wäre. Zum Beispiel können Listen aus Excel sehr einfach ersetzt und in Apps integriert werden.
Was habt ihr für Anwendungsfälle bereits umgesetzt?	Ein Anwendungsfall ist eine App für den Sicherheitsbeauftragten. Sie ist mit Microsoft gebaut und passiert über eine Liste, in welcher er Fragen beantwortet. Die Fragen lassen sich mit dem Ampel-System beantworten und mit Kommentar und Fotos ergänzen. Den Rapport kann er anschliessend Webbasiert und weiter versenden. Die Daten werden automatisch in einer Datenbank abgelegt. Dieser Anwendung fall macht ausserdem Sinn, weil im Vergleich relativ viele Unfälle passieren, was behoben werden will. Der Sicherheitsrundgang wird alle zwei Monate wiederholt.
Was habt ihr für Anwendungsfälle bereits umgesetzt?	Anwendungsfälle: Zu- oder Absagen für Kalkulation: Daten werden ab standardisierten Dokumentennamen abgegriffen. Wenn der Polier nun Daten zu den Mitarbeitenden braucht, hat er dort alle Angaben bis hin zum Temporären Mitarbeiter.
Welche Risiken haben Low-Code-Applikationen?	Zukunftsmässig muss sicher weiterhin die Daten-Durchgängigkeit besonders beachtet werden . Sie hat enorm viel Potenzial.
Was wollt ihr in Zukunft mit Low-Code umsetzen?	Wir werden auf Basis unserer Bauwerksmodell in unsere Kalkulationssoftware übertragen. Von dort aus sind wir mit einem Partner an der Lösung, dass wir E-BKP und MPK automatisierte Kalkulationen ausführen können.
Wie geht ihr mit der Datenqualität um?	Mitarbeitern vertrauen wir, dass sie die Daten richtig behandeln, weil Zugriffsrechte sonst sehr komplex werden würden.
Was sind Chancen und Herausforderungen bei der Verwendung von Low-Code?	Low Code mit Microsoft ist sehr angenehm, weil du dir über die ganze Schnittstelle innerhalb des Ökosystems von Microsoft keinen Kopf machen musst.
Mit welchen Softwares habt ihr bisher gearbeitet?	Das meiste lösen wir über Power Automate. Wir haben aber auch Visual Scripting von Appian ausprobiert. Wir arbeiten mit Grasshopper. Bonsai oder Blender Bim haben wir probiert. Oder auch Shrebrock.
Versucht ihr Prozesse zu standardisieren?	Die Erfassung der Prozesse ist dynamisch . Wenn sich Versionen oder Zugänge ändern, kann die ganze Schnittstelle nicht mehr funktionieren .
Was sind Chancen und Herausforderungen in der Verwendung von Low-Code?	Somit können sich daraus in Zukunft einmal andere vom Bedürfnisse ergeben. Wenn Prozesse oder das System sich ändert , können langjährige Produkte wieder anders aussehen. Oder auch wenn sich die ganzen Umgebungen wieder verändern, ist es mit Low-Code natürlich sehr einfach diese wieder an die neuen Begebenheiten anzupassen.
Wer erstellt Use-Cases?	Die Needs kommen von der Baustelle von den Ausführenden und werden dann vom VDC-Team umgesetzt.
Wer erstellt Low-Code-Applikationen?	VDC-Team setzt sich teilweise aus Leuten von der Branche und teilweise aus IT-Personal zusammen
Was sind Chancen und Herausforderungen in der Verwendung von Low-Code?	Ganz klar, dass Prototyping : Mit der der ganze Power Automate Geschichten bist du sehr schnell unterwegs und kannst es danach testen.
Wie geht ihr mit der Datenqualität um?	Je mehr menschliche Interaktion es gibt, desto schlechtere wird die Datenqualität .

Was sind Chancen in der Verwendung von Low-Code?	Wie im Beispiel des Sicherheitsbeauftragten ist dieser in Excel sehr versiert, deshalb macht in so einem Fall Sinn, seine Excel-Liste mit Low-Code automatisiert auszulesen und in einer App einzubinden. Somit fällt die Nachbearbeitungszeit weg und die Daten sind immer gleich und können auch im Büro eingesehen werden.
Was ist die Herausforderungen in der Verwendung von Low-Code?	Die Kombination von Microsoft zu IFC ist sehr schwierig. Die einzige Möglichkeit ist über IFC-Openshell.
Welche Plattformen verwendet ihr ausserhalb von Microsoft?	Zusätzlich haben wir bereits Erfahrungen mit Automatisierungen über UiPath gesammelt.
Welche Plattformen verwendet ihr ausserhalb von Microsoft?	Für die Modellierung von meist linienbasierten Bauwerken setzen wir beispielsweise auch Grasshopper in Kombination mit Rhino ein: Beispielsweise für die Aushubkubatur für Werkleitungen.

Anhang J Interview mit Totalunternehmer, Schweiz

Stichwortprotokoll (Schalko, 2025)

Frage	Aussage
Arbeitet ihr mit der Low-Code-Technologie?	Ich bin mir nicht sicher, ob ich Low-Code in der täglichen Arbeit verwende. In meiner früheren Zeit als ich noch mehr Modelle selbst erstellt habe, habe ich im Revit öfter mal mit Dynamo versucht und probiert.
Welche Plattformen wendet ihr an?	Letztes Jahr habe ich mir mit LLMs Codes schreiben lassen, um Modelle zu modifizieren. (LLMs wie ChatGPT oder Gemini für Code Erstellungen, um in Google Colab ausführen zu können, was BIM Anwendungsfälle angeht.) Ich bin jedoch alles andere als ein Programmierer. Nutze jedoch die einfachen «Fähigkeiten», um gewisse Arbeitsschritte zu vereinfachen. JobRouter hat Low Coding Anwendungen wo wir Absenzen und Spesen Workflows eingerichtet haben. Beekeeper haben wir seit kurzem, was auch gute Low Coding Anwendungen anbietet und wir auch verwenden.
Verwendet ihr Microsoft Power Plattform?	Klassische Low Coding gibts bei uns Power Apps von Microsoft (was wir jedoch nicht wirklich anwenden).

Anhang K Interview bei Bauleitung in einer Totalunternehmung, Schweiz

Stichwortprotokoll (Stucki, 2025)

Frage	Aussage
Verwendet ihr die Low-Code-Technologie in eurer Unternehmung?	Wir verwenden kein Low-Code
Wie werden Baufortschritte dokumentiert, seitens Bauleitung?	Ein sehr wichtiges Instrument ist das Baujournal.
Wie wird kontrolliert, wie Bauelemente gefertigt werden?	Mit der Sichtprüfung wird die Abweichung vom Werkvertrag geprüft.
Würde es der Bauleitung helfen, ein einzelnes Bauteil im Vorschritt zu dokumentieren?	Meist wird nicht mal mit Wohnungsnummern, sondern mit den verschiedenen Losen gearbeitet.
Kann ein Bauarbeiter eine App am Smartphone bedienen, oder macht es mehr Sinn, dass das der Bauleiter macht?	Es macht mehr Sinn, dass das die Bauleitung macht.

Anhang L Interview beim Totalunternehmer, Schweiz

Stichwortprotokoll (Terzic, 2025)

Frage	Aussage
Verwendet ihr die Low-Code-Technologie in eurer Unternehmung?	Ich muss dich leider enttäuschen, leider arbeite nicht mit Low-Code.

Anhang M Interview beim Elektrotechnik-Unternehmen, Schweiz

Stichwortprotokoll (Blessing & Asairatnam, 2025)

Frage	Aussage
Was habt ihr bereits umgesetzt?	Automatisierungen mit Bauwerksmodellen machen wir mit Dynamo und nicht mit Microsoft. Die Live-Daten nutzen wir im PowerBI, wo wir eine Applikation erstellt haben, mit welcher Anrufer erfasst werden. In der Liste wird erfasst wer, wann angerufen hat. Ebenfalls haben wir eine App, mit welcher der Bäckerei Bestellungen für Aperos oder Znüni aufgegeben werden kann. Oder auch eine Automatische Rechnung und eine Automatische Archivierung von Ordnern und wieder aus dem Archiv zurück.
Wer nutzt bei euch in der Unternehmung Low-Code	Wir sind eine Art IT-Team, welches unter anderem Low-Code nutzt.
Weshalb nutzt ihr Low-Code und nicht Hardcode?	Hardcode ist für uns unverständlich, wenn der Ersteller abwesend ist. Deshalb verwenden wir oft eine Mischung aus Low-Code und hardcodiert.
Wer definiert bei euch die Use-Cases?	Die Bedürfnisse von internen Kunden und fördern die Freude an der Arbeit.
Was sind Chancen und Herausforderungen im Zusammenhang mit Low-Code?	Mit Low-Code sind die Applikationen einfacher zu verstehen und anzupassen. Hard-Codiertes ist schnell unverständlich und wer hilft uns, wenn die Applikation nicht mehr geht, Aufgrund Update oder ähnlichem?
Werden die Arbeitsschritte festgehalten?	Die Arbeit wird immer dokumentiert
Stellt ihr die Applikationen immer für die ganze Firma bereit oder nur für interessierte?	Die Applikationen werden immer für ganze Unternehmer-Gruppe verwendet.
Wie werden die Applikationen in der Firma eingeführt?	Die Schulungen von Applikationen braucht viel Zeit, ist aber nötig und werden konsequent umgesetzt.
Was sehr ihr als Chance?	BIMtoField sehen wir als grosse Chance.
Was wäre eine Anwendung auf der Baustelle, welche euch helfen würde?	Eine Stückzahlen-Ermittlung der Verbauten Elemente auf der Baustelle. Darin aufrufbar eine Auswahl-Dropdown-Liste mit den Auswahlen Gewerk, Stockwerk und Anzahl und danach Exportieren.
Was sind die Chancen im Zusammenhang mit Low-Code?	Für Low-Code werden weniger sehr hoch geschultes Personal benötigt.
Was muss unbedingt beachtet werden im Zusammenhang mit Low-Code	Weitsicht ist im Management sehr wichtig, damit die richtigen Projekte verfolgt werden.
Für welche Firmengrösse und Art ist Low-Code geeignet?	Ist Low-Code nur dann etwas für grosse Firmen, wenn sie nicht in der Breite angewendet werden wollen.
Was findet ihr wichtig, Was wollt ihr mir noch mitgeben?	Es sollte wieder mehr gemeinsam für alle gearbeitet anstatt alle gegen alle.

Anhang N Interview mit Digitalen Spezialisten für das Bauwesen, Dänemark

Transkribiert mit MS Teams (Vesterlund & Falk, 2025)

Frage	Aussage
<p>What have you already done with Low-Code?</p>	<p>We've made several dashboards for instance, so we have. We use a lot of power BI and a lot of Excel for the Power BI and there we have a number of examples of that. So, we've used it for cost control, for and especially we've done a lot of dashboards on model progression. So, when you have the BIM, we take out the data and we convert it into dashboards. CDs like they look so ACC, and we pull out the file information there and we see what revisions we have and how many files we have and if they're named correctly and all that stuff uploads and whatever kind of data that you can extract so that. Type of progress data and we also do it with. We also have some code to extract from planning systems. So when we do location-based planning and they do the progress reporting in the plans. we have some small scripts that pull it out so we can stick it in in dashboards as progress reporting and do some calculations on that. And then you can compare actual versus planned. we also have some scripts that we use for tracking consumption. So we have energy consumption.</p>
<p>What have you already done with Low-Code?</p>	<p>And some small robots that go out and take information from public registers. So if you type in an address, it'll go to six or nine different websites and pull out the public information on the building. Because that's distributed to different public systems.</p>
<p>What have you already done with Low-Code?</p>	<p>It's not no code because people code a little bit, but we have low code we use for our generative design. So we do a ton of work in Grasshopper, specifically Dynamo Dynamo. Yeah, pulling information out of projects, reading a number of drawings or enriching models if we want to kind of upgrade them to be able to do a certain analysis afterwards. So we might get a model at a certain level, but we need two or three informations on the components and then we do further analysis after that. I don't think we do any actual coding anymore. Because what we'll do is we'll use ChatGPT. Or something like that to generate the Jason or the Python code and all that. Or to check our code. So it and that that also takes it from actual coding into, you know, a more simple way of doing it. But when we work with a Python, even if we're not doing it in in Grasshopper, but are working in one of the Python I can. Remember which one where PY smart I think is the one we use most. It is visual programming. That we that we use to generate the Python code and then we can push that to. A number of different platforms, even with some Revit stuff we've done, in Python.</p>
<p>When I come back, go back to something you said. Is it the progress of the components. Or more generally.</p>	<p>Now we've done both or all of it. So we it could be both the progress of the model design itself before you go on site, but it could also be the progress of the what's built compared to what was supposed to be built according to the model and the schedule compared, so it could.</p>

	<p>Be both. I can see if I can just log on to. I think I can show you some of it. Power bi.</p> <p>I think this is an example of a project where we've built some dashboards and we pulled the data out of it's. Of course in Danish, but so we pull the dates out for a number of different things from different sources.</p> <p>So this is for the models we do an extraction of the different models and we can see this is the status how many elements are in there and this is the number of collisions. And they're classified in in majors and minors. Oh, this is our favorite. You can say a table of content from the models. From the model using a script and then we have a dashboard that shows how well it's classified and coded. Lca data that we pull out of the models and.</p> <p>Then we run it through the Danish LCA calculator, and then we push the data so you can see we have different versions of that. So you can see how that is changing.</p> <p>Where the LCC calculations. We have the status of the CDE. We can see the files and if they're correctly named. Over time, there's the cost. That's taken from the estimation tool. You can see all the different estimates that are done in the project. Can you put potential compare them also? Yeah. You can see how the budget has developed. So it looks a little bit weird now, but here you can see the progress reporting. So that's we, we have a little bit little scripts that pull out the data from the progress report spreadsheet and from the plan and from the. Models and from all that, and then put it into a dashboard.</p>
<p>But you only use it for your process in in the plans, not also on the construction side.</p>	<p>the example I showed you was from the from the design period, but it could also be during the construction phase and see if I can find one.</p>
<p>What do you both?</p>	<p>Construction and what the clients normally are very excited about is that the data kinda get visual and they can easily. Decide and make decisions on whether they wanna do one or other things, because you kinda get data on the LCA and the budget and the time and what is the most. You can see we spent all the money. Some detailing of the cost versus budget. Something about the quantities, the risk profile. The model progression. And the change in cost over time. So that's a dashboard from actual construction. I'll see if I can find. you know it was a part of the project plan from the very beginning that we would do all of these analysis and visualize it as a. Power Bi project, both for the construction manager but also for the for the client. So they get an inside of the consequences of choice made.</p>
<p>But that's the way you always do your project. It's not demanded by the customers or.</p>	<p>No, it's different every time. We would love to just make a standard dashboard that we could sell to different clients, but all the projects are different. So they have a different schedule. The breakdown is different, the risks are different, the focus is different, the location is different. So. Everything is different. People say, oh, you've built all this tools for generative design. Of this, that and the other. And we have, we've done generative design on everything from wind turbines and houses and yeah office buildings and all this stuff interior design to make it into a software and go well the input is different every time the time we. Get called into. The process is different, so the preconditions are different and then the output that the clients want are different. So we can't just take format or have our</p>

	<p>standard grasshopper scripts because the input is different. So that's why we have to build the codes every time for and of course we learn something.</p> <p>So we have some you can say we have some scripting philosophies or some a code base that we can use but we can't reuse the code. We can reuse bits and pieces of it and then reconstruct them for the next project, and is that the other thing that's int we just we're having one project that we're finalizing right now where they're doing district heating and we're doing a code.</p> <p>Where you put in the endpoints and some parameters.</p> <p>And it does a number of calculations to find out what's the shortest network and how can you minimize the dimensions and all that.</p> <p>So try to optimize it based on cost. Eventually so. So, and there's a number of design criteria that goes into this. But as we progress the development of it. The designers, they put more and more rules in, but they actually can't formulate them to begin with.</p> <p>No. So we it's an iterative process, we say, OK. So it's like this and then we make a script and go, oh, but we also need this.</p>
<p>But can you use the basic or something?</p>	<p>I think we can use the philosophy sometimes to say, oh, you know, for us it doesn't matter if it's a containers on a ship deck or whether it's apartment layouts. It's the same algorithm. So the same logic that we use so it. It's a little bit of saying, oh, remember when we did this and this and or developing a specific code for.</p> <p>A specific problem, so it's. It and it is a little bit the what's great about the, the, the. The low abstraction code is that we can fairly easily make some Python or and they become quite complex quite quickly. But we can quite we can develop that. But going back and retrofitting a Python code that's made for a turbine for some district heating, it's easier to just say we know how to do this and just build on your experience and then make a new version of it. Because it's fairly fast to code. And the parametrics, I think that's also important.</p> <p>We did a parametric model of a civil project.</p> <p>And the parametric is different from project to project.</p> <p>So if you go in and look at this project and say, well, it's the width and the depth of the tunnel and it's something about the gradient of the ramps and all that. And that's completely different from other sheet piling and excavation stuff we've done when we did the excavation script for, for, for preparing, it was a different problem.</p> <p>There was also a ramp and there was, but it was more.</p> <p>There was the pocket size. Yeah, that they could remove that the time. So was the capacity of the. Oh, the trucks could enter.</p> <p>So it was the pocket size of a truck and how many trucks you could have at the same time. Yeah, and not the gradient of the ramp.</p> <p>And so it's different. A problems every time.</p>
<p>You said you always or you most of the time use low code and not hard code. Why is this?</p>	<p>It's super difficult to hard code.</p> <p>And you get high and you get you have a there's a lot of data and you can reach a quite high level.</p> <p>Yeah, with a with low code, yeah.</p> <p>It's sometimes. Sometimes we have to use hard code, but it we I think what we often will do then is do it in low code and then have chat pt transferred from the to C# or to whatever we need. The thing is also.</p> <p>If if you go to the.</p>

	<p>High abstraction languages like C# or C++ or something like that. It's more difficult to change the code.</p> <p>So if you're doing an iterative.</p> <p>Way of developing the scripts where you show something to the client and they should also be able to do this and this.</p> <p>Then it's easier to add a few more boxes and strings and parameters in in the visual low code and build like a prototype.</p> <p>And then when you end up and saying, OK, this is the script we want to run each time.</p> <p>Then the low code is often not necessarily super economical. It takes longer and it's this than the other, but then we can transfer that code into some C# that you can then run. In in in that, but we don't that often. I think often we end up with having some Python code that is in the middle of high and low code but the parameter of being able to do changes as we go along is very important because when you.</p> <p>First, show off something for the client.</p> <p>Then a ton of new ideas arises to them.</p> <p>And can we can we?</p> <p>Yes. What's the purpose of it?</p> <p>What is it that we want to achieve? So the flexibility is very important. Yeah. And I think I think it's also important that we don't try to make software, we make scripts. So we utilize, we utilize you know the backends of the software.</p> <p>So we utilize, you know.</p> <p>The back end of Power Bi of Rhino of. Revit of.</p> <p>Deluxe and all that. Because we're not actually interested in building the front ends and the software and all that because the users, they know how to use those tools.</p> <p>So we can we can make some scripts that they can go in and press some buttons inside Revit and it works better for them than it is if we build something as an application that they then have to do.</p> <p>So we kind of like.</p> <p>A like a hermit crab, we move into a existing shell.</p> <p>That, that, that that works.</p>
<p>So, what's the opportunities you think for low code?</p>	<p>I think it puts the I think you can remedy a lot of the missing functionalities in the existing software.</p> <p>Yeah, by doing it by doing these things.</p> <p>So everybody's over the last 10 years have been complaining about Revit and ACC and PIM 360 or whatever.</p> <p>And now we can say, you know, if you think it sucks, then just build your own.</p> <p>Scripts in Dynamo or in in in Rhino?</p> <p>And we haven't, we haven't run out of functionality in Rhino, there's still.</p> <p>We still have a that's not our main challenge, but you can you can. You can do more with the tools you have. Rather than having to develop your own software and I think.</p> <p>We work in a space where interoperability is super important, so being able to make a model in Revit or in rhino so it can talk with, you know, people who have MIS model in in other things. Things so we can export it to navisworks and it's much better than us making our own data structure and inventing our own.</p> <p>Making it super complicated.</p>

	<p>Yeah, that is just that doesn't work.</p> <p>No. So having something that's interoperable is much easier by scripting stuff that can transport from. We also have a lot of a lot of the companies where we do the digital landscape for help them. We're saying to just move into Microsoft World because there we can build small power apps.</p> <p>Yeah. And we can connect your business central with your excel with your this than the other and as long as you're in the Microsoft Word, somebody's updating that every 14 days. And they put millions of dollars against it.</p> <p>Yeah, and your stuff would be compatible and interoperable all the time.</p> <p>Whereas if you go out and build your own app or your own system, it's just gonna die in six months because somebody updated. We see this all the time somebody updates.</p> <p>You know the VR.</p> <p>Plug in in Firefox and then three buttons and a connects don't work anymore because.</p> <p>Yeah. So having something that's interoperable and where you can just run the scripts or just change them a little bit helps a lot.</p>
<p>What's the opposite, what's the challenges?</p>	<p>I yeah, I think it.</p> <p>Maybe it's too easy to make some code and execute on it. They can wait.</p> <p>I think we're running with a lot of prototypes, to be honest.</p> <p>Yeah, that's true.</p> <p>But that's also, yeah, it's still the advantage also that you kind of got the flexibility that you can.</p> <p>Yeah, yeah. But we were talking about the challenges.</p> <p>I think that's that backside of it.</p> <p>Yeah, it's not very well documented and it's the quality control is is debatable and the validation of the code is also.</p> <p>If we think it works, we'll continue on the way. We trust it, yeah.</p> <p>But we work a lot with prototypes, I think.</p> <p>Yeah, but it's. I think that's also important because you don't want to build software for five people.</p> <p>That's one of the things we always say to people with, they say, could we build, we have the discussion right now with a company that says can you build me a software that does this, this and this and how many people I we're six OK we would rather.</p> <p>Train you in using rhino.</p> <p>And help you build some scripts, because in 14 days and 14 months you will come back and you'll have a new idea of what the software should do. And it's actually better if we use that platform.</p> <p>That's updated every 12 months and then we can do some cool scripts that can do some stuff and then build your competences rather than build some systems.</p> <p>So if it has to scale.</p> <p>At a small scale, then the low code scripting, putting things together with Garmin duct tape.</p> <p>Super Fine because you can.</p> <p>You can do a lot of stuff quite quickly, yeah.</p> <p>And you can connect it.</p> <p>You know, we can take data from our metapore or scanners or something like that.</p>

	<p>Pull it in and do something with that because it has the flexibility, whereas if you build hard code software, it needs to be validated. It needs to be checked.</p> <p>It needs to be all these things, but it that also requires that they that you scale it, that you have 100 people using it or 1000 or 1,000,000 people using it.</p> <p>So the business case is also different.</p>
<p>it's always the same with the the outputs and the inputs and the datas who have to be.</p>	<p>Yeah, no.</p> <p>But that that's a that's a very important part of where the low code stuff and the prototyping for the situation makes sense is that if you have a project, it's very we we've done some master plan scripts for instance, yeah, but we've never found two.</p> <p>Projects where we can use the same script because the city plans are different, the requirements are different. Yeah and. the, the but the input from the client is different.</p> <p>Sometimes they don't know something, and if you if you have a rigid software like the spacemaker or former or what's it called Clifton's.</p> <p>Oh man, no, not formid.</p> <p>Oh, never mind.</p> <p>But some of these generative softwares, for early planning, you'll need to put in some, some, some data.</p> <p>And if you don't have the data, you can't progress.</p> <p>And what happens is that you just make up some ****.</p> <p>Is it?</p> <p>Let's just say it's 10,000 square meters. All of a sudden you put in a major parameter, just pulled out of the thin air because you needed that data to progress in the software.</p> <p>So it's the same when you buy something online.</p> <p>And they ask you, you know, you go on American website and they go, which state are you in?</p> <p>We don't have states in Denmark.</p> <p>Well, we have a nation. It's one state.</p> <p>But you can't select Denmark.</p> <p>Denmark. So all of a sudden you can be caught in a software. What am I going to answer here? And it's the same with the software. If you have to put in the parameters and you don't know, then you'll just put in something ***** and you get a parameter for your output. That's actually not correct.</p>
<p>It doesn't work, yes.</p>	<p>Potentially might cause very big you're optimizing on something that you just pulled out of thin air, yeah.</p>
<p>But how many people are you in your company?</p>	<p>Currently we are 8.</p>
<p>Are you managing your data quality?</p>	<p>But we are always two people on a job, so we always no one's ever working alone. So that's kind of one part of it.</p> <p>And then we, yeah, based. The next project also always on what we did last time.</p> <p>Yeah, but it's the practice.</p> <p>It's not the script or the data that we use exactly, so it's the best practice of what we have done on one, 2-3 or four different kind of projects and then we take the advantage of that to build upon the next one. Also, because we always when you've done some code three years ago and you go back and look at it, you always think it.</p>

	<p>You always remember it better than it was, and you can actually do better now, now than you could then.</p> <p>Yeah. So I think it. But there sometimes there's a logic or an input data set or something like that that you can. Let's do it like this.</p> <p>We actually had one project that we took up again.</p> <p>It had been lying dormant for almost three years and we took it up again.</p> <p>I would just restart it because.</p> <p>It was a little bit clumsy what we have done three years ago. Yeah, and parameters might have changed over the course of the three years too. So the client might have a new perspective. Absolutely has happened. The cost of the time or even the location might sometimes go wrong too.</p> <p>So all the all the parameters you want to optimize or that you can change? Yep. So.</p>
<p>How long do you work with low code?</p>	<p>How long we've done it, we've always done it. Yeah, I think we've.</p> <p>Yeah. How old is the company?</p> <p>Seven years.</p> <p>Yeah, we've done it from the beginning. Yeah. One of the first big projects we started out that was that was based on that.</p> <p>Yeah.</p> <p>No, you actually started by codifying.</p> <p>Yeah, all the local plans and all the drawings and all that. So taking things from.</p> <p>Flat in from 1D information and typing it into coordinates and making rules out of local plans and zoning plans and yeah.</p> <p>Yeah.</p>
<p>How long do you work with low code?</p>	<p>I think one of the things we found out is that you can't sell computational BIM or scripts or we've had all the.</p> <p>I think we've had one or two cases where we've sold a script over the last seven years, but you can sell.</p> <p>A classic product that's made in a different way.</p> <p>So just like, just like 3D printing. Maybe that's a good parallel.</p> <p>Maybe it's a **** parallel. We'll find out when you buy something that's three 3D printed. Not a lot of people care that it's 3D printed.</p> <p>You know, it's So what?</p> <p>But what they care about is that it's cheaper, it's lighter, it's a it's A has a, you know, the design is more valuable somehow and it and that it was produced with 3D printing. It's not that important. Well, so for all intents and purposes that that's it.</p> <p>And it's the same with the scripting.</p> <p>It's a way for us to work that we like.</p> <p>That's better for us where we can deliver something in higher quality faster and that has been you know where we've looked at hundreds of alternatives instead of two.</p> <p>But we often end up handing it over as drawings or pictures or something like that, as you would classically or as spreadsheets with numbers.</p> <p>Yeah, in an output that the that the receiver kinda recognizes.</p> <p>So it's the methodology going from the problem solving the task we're given.</p> <p>How we hand that how we process that and how we solve it and then we hand it over in PowerPoints and dashboards and as you</p>

	<p>would conventionally because the users can't necessarily take in a script. So I think we have a couple.</p> <p>Maybe it's only two or three that we have actually made scripts for that have taken over the scripts. Yeah, I think it's good.</p> <p>Yeah. And that's a whole different discussion.</p> <p>Because what's the business model for script?</p> <p>What's the business model?</p> <p>Do we have an obligation to maintain it?</p> <p>Do it. Is it a little bit like software where you have a help line and all that? Who has the IP that when we've done a script from somebody, can we use the ideas from that script another place?</p> <p>And so that's and. And the market has not decided on any of those things yet. Is it still our script or when people pay us by the hour to make it?</p>
<p>Do you think that low code can change the construction industry and the way we build.</p>	<p>Yeah, to some degree I think.</p> <p>I think at least you could have a more data valid foundation. Yeah, to make your decisions upon.</p> <p>So the number of alternatives that you can make.</p> <p>On a question or task is way higher than the human being can ever come up with. So just that perspective on it is one parameter. And so yeah, I think I think. Yeah, I would love if more people work like this because it's also more fun kind of to see the flexibility instead of just, OK. Now we did this last time we built something similar to that. So we will do almost the same once again, instead of kind of put it up open and see. What can we do? I think also the conversion of data into, you know something else to process data. And interoperable data. So if for instance, if you take machine data. Either getting the input to the machines or getting the output from the machines and turning that into, you know, financial data or progress data or something like that. It's been a huge barrier in in, in construction taking a construction program and turning it into a.nMetric design you can actually do that with low code and especially the low code. That's called the gpts. If you go over to the custom gpts you can put in the data and you can say can you analyze this data and then make some code that can transfer this into a table of a plan or execution, whatever it is so. I think that part of. Of the low code is, gonna revolutionize construction more than the people, I guess. Sitting around doing Jason and Python code. And I say this not because I don't think it has the potential, but because I don't think the industry has the interest in it. People, people, they look like a screensaver. When we normally talk about this, they go like they don't. They don't care. It's like that's fine. You do that, they don't. But they don't understand either, and they don't want to understand because they are safe on their spot.bAnd yeah, if you do something different, it's fine by me.nl just need this output. Yeah, and that's because if you're in, in, in construction, if you're in architect and engineer a contractor or subcontractor or supplier. None of those people have an interest in making scripts. That's not why they choose the career they have. So for us, we're just, we're just nerds that have found that the technology can help us do our job better. We most all of us are construction professionals. We're architects and engineers. All of us. So we have no it people or data people here. We just all found out quite early that working with the models and the data and all that made our life easier, but it is a</p>

	<p>hard sell. It is most people, they do not want to come behind the curtain. And see how we do it. It is a little bit like a restaurant. They wanna get things served on nice plates. That is exactly what they ordered. They do not even want to go behind the bar and see how the waiters work and they do not want to go into the kitchen and learn how to do potatoes. It's simply not, they just they go. You can do that. We don't give a **** and we've tried. Many, many times to see how far we can pull people back. They're not interested.</p>
<p>Yeah. Then then a little bit similar like Switzerland. thought in the in the first that that it's different in Denmark but. It seems like it's always the same.</p>	<p>I think in Denmark we're quite far with some of the digital stuff. So we are very digital nation. I think we are measured to be the most digital nation in the world. Yeah, but that's actually because of the public sector. So we have we can do our taxes on our phones. You know we can do everything digitally in Denmark. You don't need paper. Not for anything but anything, and we don't get any mail. So you don't you don't get any paper mail. No, in your mailbox don't. Nothing. Simply the local newspaper. If you sign up for it. Yeah. You don't have to sign up. No, nobody does that because they use the Facebook group since that. I literally don't check my mailbox at home. I think you will check ours here once a month or something like that. Yeah, when you go, oh, we have a mailbox. It's we get everything digitally and on projects, everything is run digitally. Yeah, on most projects, yes, they print some of the drawings and all that. But the whole manufacturing of the drawings and they use they use blue beam. We use Deluxe, we have BIM on every single project. Or at least 3D cad they use apps for hours and everything, so it's nobody puts a pen on paper anymore. Here, to be quite honest, and if you go to a meeting on construction site, people will break out their iPads or the PCs and if somebody takes out a piece of paper, you know it's a doodling. So in that aspect we have we have ICT agreements on all projects. Yeah, but that governs the exchange of information and communication and all that. Yeah, Dalux is very prominent here, but also we see Bin 360, ACC every once in a while and aconcs and all that. Yeah, it's and we have. We have 3D models of very high quality. But there's always a model behind it. So we we yeah. So in in that aspect it is very digital. They have a small group of people that work with computational BIP, so the architects and engineers, they have computational BIM people so they can get supported. Yeah, be more efficient. We were part of the network of computational BIP. Where we sit together with those companies and talk about how we work and how frustrated we are that nobody thinks is amazing. Share information and knowledge and learnings. It's always difficult to leverage digitalization in construction because it's digitalization that the value proposition. It is usually something about operational efficiency and that that's not very interesting and it's not a lot of money. The digitalization value proposition is in giving the customer a different journey. Getting access to new markets and new products, and it's about scaling something that's super cheap to scale. So if you print a drawing and you have to print a million, it's super expensive. If you have a digital drawing and you distribute it to a millimustsuper cheap. It costs the same for one person as 1,000,000 people. So that's the digital model is always in scale. Yeah. And it's always something about the product and the customer that needs to have higher value. It's not about saving money on because you know,</p>

	<p>we can save a couple of hours and a couple of people and a couple of prints, but we build things that cost millions. So, it's ***** to try to get operational efficiency out of it. It's higher value and it's scale.</p>
<p>Maybe that's what they would be interesting for a company.</p>	<p>But that's the other problem with. With digitalization and construction and all that is. The innovation model and the startup model in construction is bad. Yeah, it's and in Denmark, it's super bad. There's no venture capital. There's no, there's no funding for anything. So, every time and that means something that means that you can't put if you are in another industry, some of the companies would say, oh, but we're going to make this corner over here, where 12 people can sit and develop an app or a system. Or something for us. And then that will turn into a, a ton of money. But because it's project by project and even the big companies, when you open the lid it's just a few projects that are running there with a very thin layer. Governance and corporation. So, most people have most of the people working on projects and very few working in driving the company. So, it's very difficult to get. Systems are built in a in a company that can scale and it's very difficult to get the money. Also, because the architects and engineers and the and the contractors, historically they don't make money like Uber, and I just booked an Airbnb, and I think Airbnb they take 15%. Of the charge just for running the app. We don't have that in construction. You know if the software costs money. We put a software fee on some of our projects and said or tried and said if we go on this project, we're going to need to buy this software. And you know ACC and Autodesk stuff and rhino and all that. The clients should pay for that because they're benefiting from us, showing up with digital competence and the software and the hardware. Nobody wanted to pay for that. Nope, they'll pay for hours, but they will not pay for the tools we use, and they will not pay to get the new tools developed. So. the business model is super important. You can't get any investment. Nobody will invest in new technology. They will happily buy a new yellow machine because they can write it off for over 30 years. Yeah, but they will not. They will not buy new technology unless they must. And we need the money to develop technology, so our business model is that we go into a conventional project, and we do conventional services. But we just do them quicker and better because we have technical savvy and we try to improve every time. Yes, the work method is just different. I also want to think there's something about the mental part about this, because we see opportunities. Instead of obstacles, yeah. But we also just get super annoyed of doing the same thing twice. So that's also to, to be honest, that's also a little bit. That's also a part of it. A task that we did five years ago. We'll find a new way of doing it, because that's how our brains work, so that's why we will, you know, if somebody asks us to do something that's repetitive and boring, we'll find out a way to do a script of it. And we would rather spend the time on doing a script and debugging that. Then we would do just doing it the classical Safeway.</p>
<p>Yes, but, but that's the right way. Because nobody wants to do things twice.</p>	<p>Well, if you look at construction and you come out and propose to them, let's build this building in wood and let's do this, this and this, most of them will say I have 30 years of experience of doing it like</p>

	<p>this. Let's just do it the way I've done it, because then I know the risk and I know the time and I know what it's going to take. And that's the architects, the engineers, the contractors, the suppliers. Everybody would rather do what they've done for the last 30 years because it's safe. So that's yeah. So, the mindset that our mindset, the reason we work together is also because nobody else would want to work like this, this this way.</p> <p>Because they would all like to use their best practice and the way they've done it for 30 years, so they know how it's done. It's very few people in construction that get up every morning and say how can we do this differently, smarter, how can we invest ours?</p> <p>Brain cells just tighten the bolts a little bit and run a risk of not being able to do it on time and all that because sometimes sometimes we must pay to get jobs done. Because we overestimated our powers. But that is rare indeed. We have a few cases that cost us a lot of money. And won't even fund. So that's we're happy to pay to go to work, but then it must be fun.</p>
<p>Well, I have lots of questions, but I didn't. My English is something. It's not the best, so I can't manage it, so I just ask, have you anything else that you would like to add with? Or anything that you want to give it with me for my thesis.</p>	<p>I have one thought and that is. I think the spread of technology. It has to do with how easy it is to use, and pimp is not. If you look at the number of users across the industry, BIM is super complex. Hmm. So that's why people still want drawings, and they would just want the viewer, and you know it. It needs to be super easy, but the so the spread of BIM is limited. The spread of the CDs is bigger. More people are using it for quality assurance and all that, but that's also because it makes sense. It looks like the. The same app you have for your banking and your parking and all the other stuff. You can take a picture, and you can stick it in, and it makes sense. And I think the low code needs to tap into more of that and less of the BIM complex stuff. So, you need to recruit users that are already doing more and if you look at construction, most of the blue collars won't use the e-mail on and all that they'll use the CDE and write a note or take a picture or something like that. They are confident with that then they are with the office package, and they are with the with. And if you look at the spread of ChatGPT, I don't what it's like in Switzerland but here. Everybody uses it, so it's super adapted. So, I think putting low code in as something where they can customize the ChatGPT to do things for them is a super good entry point. For, for scaling. Low code because I think that's that. That's potentially where the biggest scaling is gone. Is that the blue colors or the people who are not sitting in front of a computer? All day but have real jobs that they will use their phone to build small ChatGPT based, or open AI based. Because that's the predominant one they have. But if you go to platform.openai.com instead of just going into the app, you can build a custom GPT in. That is where most users are, yeah. Whereas if you go into the generative design, how many people, if you look at the amount of people just in Switzerland, how many people are sitting with the design tools and? I need help to do that. But it's it that 1% of the people work in construction. 2-5 maybe 5%. At a maximum, yeah. What's it? Whereas maybe 80%</p>

	of the people that work in construction can get help to do interoperability things or fill out reports or do all sorts of things in their daily life. With if you teach them low coding. On O coding just a little bit of visual. Coding or some prompted DPT? You can help. 80% of the people are in construction.
We should give them the application they already know something like that.	Also, what is the complaint of people when you go out and talk to them, it's often that they must go to six different systems to do something, right? They have 10 different systems and to do weekly hour reports they have to go into seven different systems to do that. They can know on their own phones. Sit down and build a small app where they just say every day, they'll just talk to it. Or I've done this, this and this and it can make the data to the systems. So those types of things we can teach people how to do easily with a. With a local tool we have, and they can. They can access multiple software or data streams at once and that makes their life a lot easier and makes the data a lot better. Or they can translate data, they can say go in and read these six systems and tell me how many hours I do have left to do what on this project. So that they can manipulate the existing systems. And get relevant information for themselves. That's what people are using. That's one of the key things people are using for social media. People set up. I think the tutorial on open AI web page for the page for building AI assistance is how to build an AI assistant that draws through your social media and pushes to you what you think is interesting and not what the different algorithms thinks are int. So, we can go in and sort in all the different social media that you're in and get a thinner stream of things that interest you. It's about scale, yeah.
And how can you make their lives easier, not what others want to have because they don't want these.	You must clip your development into things that are moving. If you go to something that's monolithic and isn't moving, like design for instance, it's taken us 30 years to move from being an idea to being what it is today. That's not moving very fast, and it's only a small part of the population in construction knows how to use BIM and the models. If you look at communication and the ChatGPT and all that it's spreading out, there's multiple applications of it that people can use so that that's that part of it is running. So, we have we must look at adoption. Where? Where can it create value? Where does it get adopted? And then attach the development to those agendas.
I thank you for your Time it was very Interesting!	Thank you too. we hope we See each other again!

Anhang O Interview mit einem Digital Consulting-Unternehmen, Schweiz

Stichwortprotokoll (Oberholzer, 2025)

Frage	Aussage
Was habt ihr bereits mit Low-Code umgesetzt?	Wir haben eine Plattform, welche LCDM-Hub heisst. Dort werden Daten aus Common Data Environment aus dem Digital Twin rausgeschält und zum Beispiel in SF Facility Management überführen oder in andere. Bei dieser Plattform nehmen wir

	eben gewisse Komplexitäten heraus, so dass die Leute nachher mit Low-Code /No-Code arbeiten können.
Nutzt ihr auch andere Plattformen wie zum Beispiel Microsoft Power Platform?	Mit Microsoft Power Apps sind die Grenzen zu gross und zu wenig reif.
Was sind Herausforderungen im Zusammenhang mit Low-Code?	Und wenn du Schnittstellen im digitalen Bau hast, welche sich die ganze Zeit ändern, weil du gar nicht genau weisst, was die Bedürfnisse sind, wer es natürlich wünschenswert, dass die Bauleute selbst oder dass Datenmenschen Schnittstelle selbst machen kann. (das ist eigentlich Grundidee, dass man das Datenmanagement Team befähigt, Schnittstelle selbst zu konfigurieren)
Was ist das Risiko bei Low-Code?	Die Gefahr bei Lifecycle Data Management ist, dass die meiste Leute die Schnittstellen machen wollen, die konzentrieren sich auf eine einmalige Übernahme von Daten und, und das ist im digitalen Bauern ist das nicht so richtig, denn die Daten verändert sich laufend und das heisst man muss die Daten ganz sie können aktualisieren und zwischen System hin und herschieben.
Was muss erfüllt sein, damit Low-Code sinnvoll zur Anwendung kommen kann?	Das heisst du musst nicht nur BIMtoFM machen nicht nur vom digitalen Realisieren zu den digitalen Betrieben, sondern du musst auch wieder digital bestellen du musst eine Bestellung aufbereiten, du musst schlussendlich digital planen und das bedeutet immer, dass du die Daten rundherum schieben kannst und dass ich sag jetzt mal ist mal eine Philosophieänderung in der in der Baubranche.
Was muss in Bezug auf die Daten beachtet werden?	Der gibt es schon unzählige Datenänderungen, das heisst das, was bestellt wird und dass es nachher geliefert wird, ist vielfach gar nicht das gleiche.
Was muss beim Qualitätsmanagement beachtet werden?	Wie kann man da das Qualitätsmanagement betreiben, denn im digitalen Bauen ist es nicht so, dass die Daten per se einfach gut sind, sondern die Daten sind sagen wir jetzt mal mich die Zeit gut.
Was kann den LCDM-Hub im Vergleich zu anderen Tools?	Du kannst einfach untereinander definieren, welche Attribute du brauchst und das heisst du musst nicht wissen, wie die Schnittstelle aussieht, also du musst nicht wissen, wie es ein Hierarchisches Excel oder ist das ein Jason oder ist das das CSV oder so? Und so bekommst du dann auch deine Daten.
Was ist eine Voraussetzung?	Die Schnittstellen müssen für agile Prozesse, gemacht sein.
Was ist das Risiko bei Tools auf der Baustelle?	Derjenige der eine Wand baut, braucht die ganzen Schichten, hingegen diejenigen der die Wand später einmal putzen braucht nur die vorderste Oberfläche. Deshalb muss für jeden auch die richtige Information zur Verfügung stehen.
Wo kann Low-Code helfen die Arbeit auf der Baustelle zu vereinfachen?	Es kann dort helfen, die Daten richtig zu überführt werden müssen. Weil die auf dem Bau und die kleinen Handwerker arbeiten mit Excel, aber die können noch nicht mit IFC und Revit umgehen.
Welche Benutzer-Rollen gibt es beim LCDM-Hub?	Wir haben verschiedene User Zugänge. Je nach Unternehmen Unterschiedlich, aber beispielsweise bei Inselspital je nach Gewerk einer.
Was ist eine Chance für die Verwendung von Low-Code?	Wie können wir komplexe Programmierungen so reduzieren, dass Businessleute, dass nutzen können, was ihnen hilft.
Kann Low-Code die Art und Weise wie die Schweiz baut verändern?	Low-Code können die Bauindustrie revolutionieren. Die Schweiz muss aber zuerst von ihren eignen Standards wegkommen.

Was ist dir sonst noch wichtig und was
möchtest du mir mitgeben?

Anwendungen dürfen niemals komplex sein, sonst bringen sich
nicht viel.

Anhang P Link Prototyp

<https://apps.powerapps.com/play/e/f8c56552-83a1-e95b-bfba-452323505c49/a/f5d0036c-89f1-4ef7-bcf7-21204b936a4c?tenantId=9d1a5fc8-321e-4101-ae63-530730711ac2&hint=9c105104-5ac5-4132-8bb3-d99a311de8ee&sourcetime=1758045620797>