

Ursachen für das Scheitern von Software-Projekten

Beobachtungen aus 40 Jahren in der Praxis

1 INHALT

2	Projekt-Definition	5
2.1	Unklare Zielvorstellungen.....	5
2.2	Unklare Anforderungen	5
2.3	Unklares Konzept.....	6
2.4	Machbarkeit unzureichend geprüft	6
2.5	Stakeholder unzureichend identifiziert.....	7
2.6	Regulatorische Einschränkungen unzureichend identifiziert	8
3	Projekt-Planung	9
3.1	Anforderungen nicht komplettiert	9
3.2	Risiken unzureichend berücksichtigt	10
3.3	Abhängigkeiten unzureichend berücksichtigt	10
3.4	Unrealistische Zeit- und Budgetvorgaben.....	10
3.5	Meilensteine mangelhaft.....	11
4	Projekt-Durchführung	12
4.1	Unzureichende Informationen und Offenheit	12
4.2	Unklare Zuständigkeiten	13
4.3	Unzureichende Koordination von mehreren Teams	14
4.4	Unzureichende Testphasen und Feedback.....	15
4.5	Unzureichendes Änderungsmanagement	16
4.6	Unzureichende technische Expertise	17
4.7	Unzureichender Übergabeprozess.....	18

2 PROJEKT-DEFINITION

2.1 UNKLARE ZIELVORSTELLUNGEN

Die Zielvorstellung definiert das WAS (**Was soll erreicht werden?**) und das WARUM (**Warum ist das wichtig?**), beide zusammen geben dem Projekt Richtung und Sinn.

Unklare Zielvorstellungen verursachen:

- Unplanbarkeit
Schwierig, Meilensteine zu setzen, Ressourcen sinnvoll einzuteilen oder Prioritäten zu definieren.
- Missverständnissen im Team
Jeder füllt die Lücken in Bezug auf Was und Warum nach Gutdünken.
- Die Motivation leidet
Menschen arbeiten lieber an etwas, dessen Sinn sie verstehen.

2.2 UNKLARE ANFORDERUNGEN

Die wichtigsten Anforderungen werden kategorisiert nach **must have**, **should have**, **could have** und **won't have**. Details, die für die Erstellung des Konzepts nicht relevant sind, werden später in der Planungsphase ergänzt.

Unklare Anforderungen verursachen:

- Unplanbarkeit
Der Umfang ist unklar und ändert sich während der Durchführung.
- Die Qualität leidet
Das Projekt wird zum Flickenteppich.
- Unsicherheit im Team
Das sorgt für Frust und sinkende Motivation.
- Die Kommunikation leidet
Erhöhter Abstimmungsbedarf, Missverständnisse häufen sich.
- Vertrauensverlust
Niemand ist sich mehr sicher, ob das Projekt überhaupt auf Kurs ist.

2.3 UNKLARES KONZEPT

Das Konzept definiert **WIE** Zielvorstellungen und Anforderungen umgesetzt werden sollen und beinhaltet folgende Komponenten:

- **Lösungsansatz**
- **Integration in die Systemarchitektur**
- **Anbindung an Geschäftsprozesse**
- **Architektur der Lösung**
- **Wichtige prozessuale und technische Eckpunkte**
- **Hauptrisiken**
- **Begründung wichtiger Entscheidungen**
- **Grobe Zeit- und Kostenschätzung**

Ein unklares Konzept verursacht:

- **Unplanbarkeit**
Ohne klares Konzept kann das Ergebnis der Planung nur eine Sammlung frommer Wünsche sein.
- **Die Qualität leidet**
Erhöhtes Risiko von Fehlern, Inkonsistenzen oder einem Endprodukt, das nicht richtig funktioniert oder den Bedürfnissen nicht entspricht.
- **Orientierungslosigkeit im Team**
Jeder macht sein eigenes Ding, widersprüchliche Ansätze werden verfolgt.
- **Ineffizienz**
Herumprobieren statt gezielten Vorgehens.
- **Die Entscheidungsfindung leidet**
Es fehlt die Grundlage, um zwischen Alternativen abzuwägen oder Prioritäten zu setzen.

2.4 MACHBARKEIT UNZUREICHEND GEPRÜFT

Die Aufgabe der Machbarkeitsüberprüfung ist es, die Realisierbarkeit eines Projekts systematisch nach den Kategorien **Technisch, Wirtschaftlich, Rechtlich, Organisatorisch und Zeitlich** zu bewerten.

Eine unzureichend geprüfte Machbarkeit verursacht:

- **Verzögerungen**
Es können während der Durchführung Hindernisse auftreten, die vorher nicht erkannt wurden (z. B. technische Grenzen). Das führt zu Verzögerungen, da Probleme erst nachträglich gelöst werden müssen.
- **Qualitätsmängel**
Wenn die Machbarkeit technisch nicht ausreichend geprüft wurde, könnte das Endergebnis nicht den Anforderungen entsprechen. Zum Beispiel könnte eine Software nicht skalierbar sein oder nicht den Sicherheitsstandards genügen.
- **Projektabbruch**
Im schlimmsten Fall stellt sich heraus, dass das Projekt gar nicht umsetzbar. Das führt zum vollständigen Scheitern, nachdem bereits Zeit und Geld investiert wurden.

2.5 STAKEHOLDER UNZUREICHEND IDENTIFIZIERT

Die Identifizierung der Stakeholder hat eine zentrale Aufgabe: Sie legt den **Grundstein** für eine effektive Kommunikation, Zusammenarbeit und das Management von Erwartungen während des gesamten Projektverlaufs.

Unzureichend identifizierte Stakeholder verursachen:

- **Fehlende Unterstützung oder Widerstand**
Nicht identifizierte Stakeholder – wie Entscheidungsträger, Betroffene oder Einflussnehmer – könnten das Projekt blockieren, wenn ihre Interessen nicht berücksichtigt wurden.
- **Unklare oder widersprüchliche Anforderungen**
Wenn wichtige Stakeholder nicht einbezogen werden, fehlen deren Perspektiven bei der Ziel- und Anforderungsdefinition. Das kann dazu führen, dass das Projekt am Ende nicht die Bedürfnisse aller Beteiligten erfüllt oder nachträgliche Änderungen nötig werden.
- **Kommunikationsprobleme**
Ohne eine vollständige Stakeholder-Liste weiß das Projektteam nicht, wen es informieren oder einbinden muss. Das kann zu Missverständnissen, Gerüchten oder gar Konflikten führen, weil wichtige Parteien im Dunkeln gelassen werden.
- **Verzögerungen**
Wenn ein übersehener Stakeholder später ins Spiel kommt (z. B. ein Investor oder eine Abteilung), können neue Forderungen oder Einsprüche den Zeitplan durcheinanderbringen. Das erfordert oft Rückschritte oder zusätzliche Planungen.
- **Erhöhte Kosten**
Nachtragende Anpassungen, um übersehene Stakeholder zufriedenzustellen, treiben die Kosten in die Höhe. Beispielsweise könnte ein Kunde, der nicht früh eingebunden wurde, teure Änderungen am Produkt verlangen.

2.6 REGULATORISCHE EINSCHRÄNKUNGEN UNZUREICHEND IDENTIFIZIERT

Folgende Bereiche sind derzeit für Software-Projekte **besonders relevant**:

- **Datenschutz und Datensicherheit**
Regelwerke wie die DSGVO in der EU legen strenge Vorgaben fest, wie personenbezogene Daten gesammelt, gespeichert und verarbeitet werden dürfen. Wenn Software nicht konform ist – etwa durch fehlende Einwilligungen, unzureichende Verschlüsselung oder keinen klaren Löschmechanismus, drohen hohe Bußgelder und rechtliche Schritte. Das kann die Entwicklung verzögern, weil zusätzliche Sicherheitsfeatures eingebaut werden müssen.
- **Branchenspezifische Vorschriften bzw. Zertifizierungen**
In regulierten Sektoren wie Gesundheitswesen (z. B. HIPAA in den USA), Finanzwesen (z. B. PCI DSS für Zahlungsdaten) oder Luftfahrt gibt es spezifische Standards. Eine medizinische App muss etwa nachweisen, dass sie sicher und zuverlässig ist, was Zertifizierungen und Tests erfordert. Missachtung kann die Markteinführung blockieren oder Rückrufe nach sich ziehen.
- **Verwendung von Künstlicher Intelligenz**
Die strengsten aktuellen und geplanten Einschränkungen finden sich in der EU mit dem AI-Act, der insbesondere hochriskante und ethisch problematische Anwendungen reguliert. Während Verbote bereits ab 2025 greifen, haben Unternehmen bei Hochrisiko-Systemen bis 2027 Zeit, sich anzupassen. Wenn KI nicht konform eingesetzt wird, drohen hohe Bußgelder und rechtliche Schritte.
- **Barrierefreiheit**
Gesetze wie die WCAG (Web Content Accessibility Guidelines) oder nationale Vorschriften (z. B. Barrierefreie-Informationstechnik-Verordnung in Deutschland) verlangen, dass Software für Menschen mit Behinderungen zugänglich ist. Fehlende Umsetzung kann die Nutzung durch öffentliche Stellen verbieten und Klagen auslösen, was nachträgliche Anpassungen erzwingt.
- **Exportkontrollen**
Wenn Software kryptografische Funktionen enthält oder in bestimmten Ländern eingesetzt wird, können Exportbeschränkungen gelten (z. B. ITAR oder EAR in den USA). Eine Nichtbeachtung könnte den Vertrieb einschränken und rechtliche Konsequenzen haben, was den Projektumfang oder die Zielmärkte plötzlich verkleinert.

3 PROJEKT-PLANUNG

Alle in diesem Kapitel angeführten Punkte verursachen gleichermaßen:

- Das Vertrauen in die Führung leidet
Ziele oder Strategie des Projekts werden angezweifelt, was die Ausrichtung des Teams erschwert.
Teammitglieder die das Gefühl haben, dass die Führung inkompetent, unzuverlässig oder nicht transparent ist, verlieren die Motivation, sich voll einzusetzen.
Spannungen zwischen Team und Führung führen zu passivem Widerstand.
Talentierte Teammitglieder könnten das Projekt oder sogar die Organisation verlassen.
- Die Teamdynamik leidet
Schlechtere Kommunikation, geringere Produktivität, sinkende Kreativität und eine höhere Fehlerrate.

3.1 ANFORDERUNGEN NICHT KOMPLETTIERT

Das Konzept beinhaltet typischerweise nur die wichtigsten Anforderungen, daher müssen die funktionalen

Eingaben/Ausgaben
Berechnungen
Interaktionen
Prozesse

und nicht-funktionalen

Leistung
Sicherheit
Benutzbarkeit
Ästhetik
Reaktionsfähigkeit
Barrierefreiheit
Skalierbarkeit
Verfügbarkeit

Anforderungen vor Beginn der Durchführung ergänzt bzw. detailliert werden.

Unvollständige Anforderungen verursachen:

- Der Zeit- bzw. Kostenrahmen wird gesprengt
- Architektur- und andere technische Entscheidungen werden falsch getroffen
- Technische Schulden entstehen
- Die Produkt-Lebensdauer wird verkürzt
- Die Produktqualität leidet

3.2 RISIKEN UNZUREICHEND BERÜCKSICHTIGT

Eine vorausschauende Identifikation und Handhabung von Risiken verhindern, dass **unerwartete Probleme** das Projekt aus der Bahn werfen.

Unzureichend berücksichtigte Risiken verursachen:

- Der Zeit- bzw. Kostenrahmen wird gesprengt
- Die Produktqualität leidet

3.3 ABHÄNGIGKEITEN UNZUREICHEND BERÜCKSICHTIGT

Abhängigkeiten – also die **Beziehungen zwischen Aufgaben, Ressourcen oder Prozessen**, die aufeinander angewiesen sind – sind ein Kernbestandteil der jeder Projektplanung.

Unzureichend berücksichtigte Abhängigkeiten verursachen:

- Unvorhergesehene Verzögerungen bzw. Stillstände
- Mehraufwände für Provisorien
- Die Produktqualität leidet

3.4 UNREALISTISCHE ZEIT- UND BUDGETVORGABEN

Zeit- und Budgetvorgaben dienen bei einem Projekt als **zentrale Steuerungsinstrumente**, um den Rahmen, die Effizienz und den Erfolg sicherzustellen.

Unrealistische Zeit- und Budgetvorgaben verursachen:

- Stress und Überlastung
- Die Planung muss ständig korrigiert werden
- Der Zeit- bzw. Kostenrahmen wird gesprengt
- Die Produktqualität leidet

3.5 MEILENSTEINE MANGELHAFT

Meilensteine dienen als wichtige Orientierungspunkte, um den Fortschritt zu **strukturieren**, zu **messen** und zu **steuern**. Sie sind klar definierte Etappen oder Ereignisse, die einen bedeutenden Abschnitt im Projektverlauf markieren.

Mangelhafte Meilensteine verursachen:

- **Unvorhergesehene Verzögerungen**
Schlecht definiert oder unrealistisch gesetzt, kann es passieren, dass Aufgaben nicht rechtzeitig abgeschlossen werden. Das führt zu Verschiebungen und kann den gesamten Projektabschluss gefährden.
- **Ressourcenprobleme**
Teams könnten überlastet werden, weil sie nicht wissen, wann welche Ressourcen benötigt werden, oder Ressourcen bleiben ungenutzt.
- **Mangelnde Übersicht und Kontrolle**
Ohne klare Anhaltspunkte wird es schwierig, Abweichungen frühzeitig zu erkennen und gegenzusteuern. Das erhöht das Risiko, dass Probleme erst spät sichtbar werden, wenn sie schwerer zu beheben sind.
- **Motivationsverlust im Team**
Teams brauchen erreichbare Zwischenziele, um motiviert zu bleiben. Sind Meilensteine vage oder unrealistisch, kann das zu Frustration führen, weil der Erfolg nicht greifbar ist oder die Arbeit endlos erscheint.
- **Die Produktqualität leidet**
Wenn Meilensteine nicht klar mit Qualitätsstandards verknüpft sind, könnten wichtige Prüfungen oder Tests übersehen werden.
- **Der Zeit- bzw. Kostenrahmen wird gesprengt**
Zeitverzögerungen, Nacharbeiten oder ineffiziente Ressourcennutzung treiben die Kosten in die Höhe.

4 PROJEKT-DURCHFÜHRUNG

4.1 UNZUREICHENDE INFORMATIONEN UND OFFENHEIT

Ziele, Anforderungen, Konzept und **Zeitvorgaben** können nur dann **von allen mitgetragen** werden, wenn Fragen dazu willkommen sind, Entscheidungen begründet werden und auf Zweifel und Bedenken eingegangen wird.

Unzureichende Informationen und Offenheit verursachen:

- **Missverständnisse und Fehlinterpretationen**
Es wird an Dingen gearbeitet, die das Unternehmen gar nicht will.
- **Inkonsistente Prioritäten**
Jeder Beteiligte priorisiert nach eigenem Ermessen.
- **Verzögerungen im Projekt**
Es entstehen unnötige Rückfragen, Nachbesserungen oder gar Stillstand.
- **Scope Creep**
Es schleichen sich leicht zusätzliche Anforderungen ein. Teammitglieder oder Stakeholder können eigene Ideen einbringen, weil sie die Grenzen nicht kennen.
- **Die Produktqualität leidet**
Wenn das Konzept (z. B. technische Architektur oder Design) nicht einheitlich verstanden wird, entstehen Inkonsistenzen in der Software – etwa eine Benutzeroberfläche, die nicht zum Rest passt, oder eine Funktion, die nicht mit dem System harmoniert.
- **Verlust von Vertrauen**
Unklare Kommunikation erzeugt Frust. Das Unternehmen könnte dem IT-Dienstleister Inkompetenz vorwerfen, während der IT-Dienstleister den Kunden für mangelnde Klarheit kritisiert.
- **Ineffiziente Ressourcennutzung**
Teams verschwenden Zeit mit falschen Annahmen oder doppelter Arbeit, weil sie nicht auf dem gleichen Stand sind. Ein Entwickler könnte z. B. eine Funktion programmieren, die ein anderer schon anders gelöst hat.

4.2 UNKLARE ZUSTÄNDIGKEITEN

Klare definierte Zuständigkeiten dienen dazu, Verantwortlichkeiten eindeutig zuzuweisen und die **Effizienz** und **Koordination** sicherzustellen.

Unklare Zuständigkeiten verursachen:

- Chaos und Ineffizienz
Doppelarbeit, Aufgaben bleiben liegen, Lücken in der Ausführung, Probleme werden verschleppt
- Die Kommunikation leidet
Fragen werden nicht geklärt, Entscheidungen verschleppt, Informationen nicht weitergegeben
- Die Teamdynamik leidet
Unklare Zuständigkeiten führen oft zu Schuldzuweisungen oder Streit.

4.3 UNZUREICHENDE KOMMUNIKATION UND ABSTIMMUNG

Feste Kommunikationswege und -termine sind essenziell, denn **Transparenz** über **Fortschritte**, **Herausforderungen** und **offene Punkte** gehören zu den Grundpfeilern beim Projektmanagement.

Unzureichende Kommunikation und Abstimmung verursachen:

- Verwirrung
- Fehlentwicklungen
- Verzögerungen
- Die Stimmung verschlechtert sich

4.4 UNZUREICHENDE KOORDINATION VON MEHREREN TEAMS

Wenn an einem Projekt mehrere Teams arbeiten, hängen **Erfolg** und **Effizienz** stark von einer gut abgestimmten Zusammenarbeit ab.

Unzureichende Koordination von mehreren Teams verursacht:

- **Fehlende Abstimmung an den Schnittstellen**
Selbst wenn Schnittstellen klar definiert sind, können Missverständnisse über Details (z. B. Datenformate, Protokolle) entstehen. Wenn ein Team eine Schnittstelle anders implementiert als erwartet, funktioniert die Integration nicht.
- **Verborgene Abhängigkeiten**
Manchmal gibt es unerwartete Wechselwirkungen zwischen Modulen, die über die Schnittstellen hinausgehen. Wenn diese nicht erkannt werden, kommt es zu Fehlern, sobald die Teile zusammengeführt werden.
- **Verzögerungen durch Wartezeiten**
Wenn ein Team auf die Fertigstellung einer Schnittstelle durch ein anderes Team angewiesen ist, können Verzögerungen bei einem Team den gesamten Zeitplan durcheinanderbringen.
- **Mangelnde Gesamtsicht**
Teams, die sich nur auf ihre eigenen Bereiche konzentrieren, könnten das große Ganze aus den Augen verlieren. Das kann dazu führen, dass die Software zwar technisch funktioniert, aber die Zielvorstellungen („Was“ und „Warum“) nicht optimal erfüllt.
- **Qualitätsunterschiede**
Wenn jedes Team eigene Standards für Tests oder Code-Qualität hat, könnten Schwächen in einem Modul (z. B. mangelnde Robustheit) die gesamte Software beeinträchtigen, sobald sie über Schnittstellen verbunden wird.
- **Kommunikationslücken**
Die klare Trennung kann dazu führen, dass Teams zu wenig miteinander sprechen. Änderungen an einer Schnittstelle werden vielleicht nicht rechtzeitig kommuniziert, was Nacharbeit erfordert.
- **Integrationsprobleme**
Auch bei gut definierten Schnittstellen kann die Zusammenführung der Teile scheitern, wenn Annahmen über Performance, Skalierbarkeit oder Fehlerbehandlung nicht abgestimmt wurden. Das wird oft erst spät sichtbar.
- **Scope Creep an Schnittstellen**
Ein Team könnte eine Schnittstelle erweitern, um eigene Bedürfnisse zu erfüllen, was den Umfang unbemerkt ausdehnt und andere Teams zwingt, sich anzupassen.
- **Fehlende Flexibilität**
Zu starre Schnittstellen können Anpassungen erschweren, wenn sich Anforderungen ändern. Teams könnten sich hinter „Das ist nicht unser Bereich“ verschanzen, statt lösungsorientiert zu handeln.
- **Kosten für späte Fehlerbehebung**
Probleme, die erst bei der Integration auffallen (z. B. inkompatible Versionen), sind teurer zu beheben als während der Entwicklung, weil sie mehrere Teams betreffen.

4.5 UNZUREICHENDE TESTPHASEN UND FEEDBACK

Es muss die Möglichkeit bestehen, **Zwischenstände** zu **testen** (z. B. Prototypen oder Alphaversionen) und **Feedback** zu **geben**, wenn möglich gibt es **Zwischenabnahmen**.

Versionen müssen eine Versionsbeschreibung enthalten (was enthalten, was nicht enthalten, behobene und bekannte Fehler).

Der Umgang mit Fehlern und Abweichungen von den Anforderungen muss einem definierten Prozess folgen (Erfassen, Nachtesten, Schließen).

Unzureichende Testphasen und Feedback verursachen:

- Fehlerhafte oder unvollständige Software
- Abweichung von den Anforderungen
- Hohe Nachbesserungskosten
- Schlechtes Benutzererlebnis
- Verpasste Optimierungschancen
- Frust und Konflikte

4.6 UNZUREICHENDES ÄNDERUNGSMANAGEMENT

Softwareprojekte sind oft dynamisch – Kundenanforderungen ändern sich, neue Technologien kommen hinzu oder Fehler werden entdeckt.

Das Änderungsmanagement hat die Aufgabe, Änderungen an Anforderungen, Prozessen oder Ressourcen **strukturiert, kontrolliert** und **effizient** zu handhaben.

Unzureichendes Änderungsmanagement verursacht:

- **Unkontrollierte Änderungen**
Ohne klaren Prozess werden Änderungswünsche spontan eingereicht und umgesetzt, ohne deren Auswirkungen zu prüfen. Das kann bestehende Funktionen beschädigen oder zu Inkonsistenzen in der Software führen.
- **Kosten- und Zeitüberschreitungen**
Häufige, schlecht dokumentierte Änderungen erhöhen den Aufwand für den IT-Dienstleister, was das Budget sprengt und den Zeitplan verzögert. Ohne festgelegte Prioritäten werden auch unwichtige Anpassungen umgesetzt, die Ressourcen binden.
- **Die Produktqualität leidet**
Wenn Änderungen nicht systematisch getestet werden, entstehen neue Fehler oder alte Bugs werden reaktiviert. Das gefährdet die Stabilität und Zuverlässigkeit der Software.
- **Missverständnisse und Konflikte**
Ohne klaren Kommunikationsweg (z. B. wer Änderungen genehmigt) können Unternehmen und IT-Dienstleister unterschiedliche Erwartungen haben. Das führt zu Streit darüber, was vereinbart war und wer für Mehrkosten aufkommt.
- **Verlust der Zielvorstellungen**
Mangelhaftes Änderungsmanagement kann dazu führen, dass die ursprünglichen Ziele (das „Was“ und „Warum“) aus den Augen verloren gehen. Die Software entwickelt sich in eine Richtung, die nicht mehr den eigentlichen Bedürfnissen entspricht.
- **Fehlende Nachvollziehbarkeit**
Wenn Änderungen nicht dokumentiert werden, weiß später niemand mehr, warum oder wann etwas angepasst wurde. Das erschwert Wartung, Fehlerbehebung oder zukünftige Erweiterungen.
- **Frustration bei Beteiligten**
Ein chaotischer Änderungsprozess sorgt für Stress – beim Unternehmen, weil gewünschte Anpassungen nicht schnell genug kommen, und beim IT-Dienstleister, weil ständig neue Anforderungen auftauchen. Das schadet der Zusammenarbeit.
- **Risiko von Scope Creep**
Ohne klare Grenzen und Abstimmung schleichen sich immer mehr Anforderungen ein („Können wir das nicht auch noch machen?“). Das Projekt wird unübersichtlich und droht nie fertig zu werden.

4.7 UNZUREICHENDE TECHNISCHE EXPERTISE

Die Grundlage für eine realistische Planung, effektive Kommunikation und den Gesamterfolg ist eine ausreichende technische Expertise beim Auftraggeber.

Unzureichende technische Expertise verursacht:

- **Kommunikationsprobleme**
Technische Diskussionen mit dem IT-Dienstleister werden schwierig, weil Fachbegriffe, Prozesse oder Prioritäten nicht ausreichend verstanden werden. Das kann den Austausch verlangsamen und Missverständnisse häufen.
- **Schlechtes Feedback**
In Testphasen oder bei der Bewertung von Zwischenständen werden technische Schwächen oder Fehler nicht erkannt oder benannt. Das verzögert Korrekturen und lässt Probleme ungelöst.
- **Unrealistische Erwartungen**
Zeitaufwand, Kosten oder Machbarkeit werden falsch eingeschätzt. Zum Beispiel könnten unmögliche Funktionen gefordert oder technische Grenzen ignoriert werden, was zu Konflikten mit dem IT-Dienstleister führt.
- **Verzögerungen**
Technische Entscheidungen können nicht getroffen oder abgenommen werden (z. B. bei Architektur oder Schnittstellen). Der IT-Dienstleister wartet auf Freigaben, während Deadlines verstreichen.
- **Fehlentscheidungen**
Es werden Lösungen akzeptiert, die langfristig unpraktikabel sind (z. B. schlechte Skalierbarkeit oder hohe Wartungskosten) oder weil die Konsequenzen nicht erkannt werden (bei APIs, Datenbanken, Middleware, Hardware-Schnittstellen, Datei-Formate, Cloud-Integration usw.).
- **Verlust von Kontrolle**
Der IT-Dienstleister gewinnt die Oberhand und setzt eigene Prioritäten durch, weil nicht genug Standing vorhanden ist, um kritisch nachzuhaken oder Alternativen vorzuschlagen.

4.8 UNZUREICHENDER ÜBERGABEPROZESS

Nach Fertigstellung sollte das Unternehmen geschult werden (z. B. Mitarbeiter oder IT-Abteilung) und eine saubere Übergabe inklusive Dokumentation und ggf. Wartungsvereinbarungen erfolgen.

Ein unzureichender Übergabeprozess verursacht:

- **Wissenslücken beim Unternehmen**
Wenn der IT-Dienstleister keine ausreichende Dokumentation oder Schulung bereitstellt, weiß das Unternehmen (z. B. IT-Team oder Endnutzer) nicht, wie die Software funktioniert, gewartet oder angepasst wird. Das führt zu Unsicherheit und ineffizienter Nutzung.
- **Probleme bei Betrieb und Wartung**
Ohne klaren Übergabeprozess – etwa mit technischer Dokumentation, Quellcode-Zugang oder Anleitungen – kann das Unternehmen die Software nicht selbst betreiben, aktualisieren oder Fehler beheben.
- **Verzögerungen beim Produktivstart**
Wenn Mitarbeiter nicht geschult sind oder wichtige Informationen (z. B. Zugangsdaten, Konfigurationen) fehlen, kann die Software nicht zeitnah in den Betrieb gehen. Das gefährdet Projektpläne und Geschäftsprozesse.
- **Übersehene Qualitätsmängel**
Ein hastiger oder unvollständiger Übergabeprozess kann dazu führen, dass letzte Fehler oder Unstimmigkeiten nicht bemerkt werden. Das Unternehmen entdeckt diese erst im Einsatz, wenn es teurer wird, sie zu beheben.
- **Sicherheitsrisiken**
Ohne eine ordnungsgemäße Übergabe könnten Sicherheitsaspekte (z. B. Passwörter, Verschlüsselungseinstellungen) unklar bleiben oder falsch übernommen werden, was die Software angreifbar macht.
- **Kostensteigerungen**
Fehlende Klarheit bei der Übergabe zwingt das Unternehmen oft, den IT-Dienstleister für Nachbesserungen oder zusätzliche Unterstützung erneut zu beauftragen. Das treibt die Gesamtkosten in die Höhe.
- **Frustration bei Mitarbeitern**
Wenn die Software nicht richtig erklärt wird oder technische Probleme auftreten, ohne dass das Team sie lösen kann, sinkt die Motivation und Akzeptanz im Unternehmen.
- **Rechtliche Unsicherheiten**
Ohne klare Übergabevereinbarungen (z. B. zu Lizenzrechten, Garantien oder Support) könnten Streitigkeiten mit dem IT-Dienstleister entstehen, etwa wer für spätere Probleme verantwortlich ist.