

GATE5-Studienüberblick:

Die häufigsten Gründe, warum IT-Projekte scheitern

Überblick: Warum wir auf die Suche gegangen sind

Was sind die Gründe, weshalb so viele IT-Projekte scheitern? Und was bedeutet “scheitern” genau?

Wir wissen: Darüber redet niemand gern. Trotzdem haben viele unserer Kunden schon schmerzhaft Erfahrungen damit gemacht, wenn IT-Projekte nicht wie vereinbart abgeschlossen wurden und sich im Nachhinein als teurer oder langwieriger herausgestellt haben. Andere Projekte werden erst gar nicht abgeschlossen. Dass das so ist, wissen wir alle. Beim “Warum” gehen die Meinungen auseinander. Die häufigste Erklärung, die wir hören ist: Es liegt eben an der Technologie.

Als Techniker hat uns das sehr verwundert. Ist doch die Technologie der planbare Part, und erfahrene Entwickler können einschätzen, welcher Aufwand auf sie zukommt, wenn sie mit einem Projekt beginnen, das gut spezifiziert wurde.

Deshalb haben wir uns entschieden zu sichten, was die Literatur zum Thema hergibt und spannende Ergebnisse gefunden!

Natürlich bleiben wir nicht beim Benennen der Ursachen, sondern zeigen auf, wie man das Scheitern von Projekten vermeiden kann, und worauf es von Kunden- aber auch Entwicklerseite zu achten gilt, damit Projekte erfolgreich abgeschlossen werden, im Budget bleiben und in der geplanten Zeit.

Sind IT-Projekte besonders schwierig?

IT-Projekte gelten als ein schwieriges Unterfangen. Die bekannteste Quelle, die sich mit der Erfolgsquote von IT-Projekten befasst, sind die Studien der Standish Group International aus den Jahren 1994 bis heute. Dabei wurden inzwischen mehrere Zehntausend IT-Projekte umfassend ausgewertet. Das Ergebnis: unter 19 % der Projekte wurden rechtzeitig und innerhalb des veranschlagten Budgets abgeschlossen. Dagegen wurden etwa 32 % der Projekte vor ihrem Abschluss abgebrochen, während die verbleibenden Projekte mit höheren Kosten als den ursprünglichen Schätzungen verbunden waren oder deutlich längere Entwicklungszeiten benötigten als ursprünglich veranschlagt (Standish Group, 2022). Interessant: Die Zahlen variieren über die Jahre nur unwesentlich. Es scheint also seit dem Jahr 1994 keine erheblichen Fortschritte zu geben, was den erwartungsgemäßen Abschluss von IT-Projekten angeht, obwohl die Technologien seitdem große Fortschritte gemacht haben.

Die Ergebnisse der Standish Group sind nicht unumstritten. Die Reaktionen lauten auf der einen Seite: Es gibt mehr erfolgreiche Projekte als behauptet. Auf der anderen Seite: Es sind noch weniger, denn einige Projekte würden, einmal abgeschlossen, "schön" gelogen. Wir sind der Auffassung. Selbst wenn die Wahrheit ein paar Prozentpunkte höher oder niedriger liegt: Mit so vielen ganz gescheiterten oder nur teilweise zufriedenstellend abgeschlossenen Projekten kann niemand zufrieden sein!

Eine 2003 von der Oxford University und Computer Weekly durchgeführte Studie kam darüber hinaus zu ähnlichen Ergebnissen wie die Befragungen der Standish Group. Auch hier konnten nur 16 % der untersuchten IT-Projekte als erfolgreich angesehen werden (Sauer & Cuthbertson, 2003).

Über die Ursachen des Scheiterns sagen die unterschiedlichen Untersuchungen aber nur wenig Brauchbares aus. Deshalb gehen wir ihnen in diesem Artikel auf den Grund.

Im nächsten Abschnitt stellen wir die Ergebnisse verschiedener wissenschaftlicher Arbeiten gegenüber, um eine Gesamtsicht über die Ursachen gescheiterter IT-Projekte zu geben. Dabei gilt es anzumerken, dass hierbei kein einheitlicher Konsens existiert, wie das Scheitern und der Erfolg von IT-Projekten zu definieren ist. Der Standish Group nach zeichnet sich ein erfolgreiches Projekt dadurch aus, dass es innerhalb des zugewiesenen Budgets, innerhalb der Frist und mit vollständiger Bereitstellung der erforderlichen Funktionalitäten abgeschlossen

wird. Wir finden diese Definition gut, sie entspricht dem, was Kunden in der Regel als “Erfolg” definieren.

Ergebnisse aus 8 Quellen

Die folgende Übersichtstabelle zeigt eine Auswahl an Arbeiten von Einzelpersonen, Gruppen und Institutionen zur Ursache für das Scheitern von IT-Projekten. Nach der Auflistung der häufigsten Ursachen für das Scheitern von IT-Projekten haben wir daraus ein Rahmenwerk entwickelt, das bei der erfolgreichen Durchführung von IT-Projekten helfen soll. Unsere Arbeit befasst sich mit Sekundärdaten. 8 verschiedene Quellen verwenden wir für unsere Übersicht.

Publikation	Gründe für das Scheitern It. Publikation
Collaborating on project success Johnson, J., et al. (2001)	<ol style="list-style-type: none"> 1. Mangelnde Unterstützung seitens der Führungsebene 2. Fehlende Beteiligung der Endanwender (User) 3. Einsatz von unerfahrenen Projektmanager 4. Unklare Geschäftsziele die mit dem Projekt verbunden sind 5. Zu großer initialer Projektscope 6. Entwicklungsaufwände auf Infrastrukturebene, anstelle Standardlösungen zu verwenden 7. Umsetzung zu komplexer Features (anstelle mit einer Basisausprägung zu starten) 8. Fehlende Methodik im Projektmanagement 9. Unzutreffende, initiale Aufwandsschätzungen
Why web projects fail Fichter, D. (2003)	<ol style="list-style-type: none"> 1. Mangelnde Planung, oft bedingt durch Zeitdruck 2. Unklare Ziele und Aufgaben

	<ol style="list-style-type: none"> 3. Veränderung von Zielen während der Projektumsetzung 4. Unrealistische Zeit- oder Ressourcenschätzungen 5. Mangelnde Unterstützung seitens der Führungsebene 6. Fehlende frühzeitige Anwenderbeteiligung 7. Unzureichende Kommunikation, besonders bei großen Projekten 8. Unpassende Fähigkeiten im Team
<p>Managing Successful Software Development Projects</p> <p>Mike Thibado(2006)</p>	<ol style="list-style-type: none"> 1. Unzureichend spezifizierte Projektziele 2. Schlechte Planung und Schätzung von Zeit- und Ressourcenbedarf 3. Einsatz neuer Technologie in der Organisation 4. Unzureichende oder fehlende Projektmanagement-Methode 5. Unzureichende Anzahl an erfahrenen Führungskräften im Team
<p>Why it projects fail</p> <p>Al Neimat, Taimour 2005</p>	<ol style="list-style-type: none"> 1. Schlechte Projektplanung 2. Unklare Ziele und Vorgaben 3. Änderung der Ziele während der Projektumsetzung 4. Unrealistische Zeit- oder Ressourcenschätzungen 5. Mangelnde Unterstützung durch die Geschäftsführung und fehlende Einbeziehung der Nutzer 6. Unzureichende Kommunikation und fehlende Zusammenarbeit im Team 7. Fehlende oder ungeeignete Fähigkeiten im Projektteam

<p>Identifying software project risks</p> <p>Schmidt, Lyytinen, Keil & Cule, 2001</p>	<ol style="list-style-type: none"> 1. Unzureichende Projektunterstützung seitens des Top-Managements 2. Missverständnisse hinsichtlich der tatsächlichen Anwenderanforderungen 3. Unzureichende Bewältigung von Projektveränderungen 4. Mangelnde Integration von frühzeitigem Benutzerfeedback 5. Veränderung von Projektumfang und Zielen 6. Komplexität infolge dem Zusammenspiel mehrerer organisatorischer Einheiten 7. Unklare/missverstandene Ziele und Umfang 8. Fehlende klare Definitionen von Rollen und Verantwortlichkeiten 9. Fehlen festgelegter Anforderungen 10. Einführung neuer Technologien 11. Mangel an effektiven Projektmanagement-Fähigkeiten 12. Fehlende effektive Projektmanagementmethode 13. Mangel an erforderlichem Teamwissen und spezifischen Skills
<p>Why Projects Fail: Avoiding the Classic Pitfalls</p> <p>Isfahani, 2010</p>	<ol style="list-style-type: none"> 1. Fehlendes oder mangelhaftes Risikomanagement 2. Fehlende Transparenz über den Projektfortschritt während der Umsetzung (Performance-Messung) 3. Unzureichende Definition von Projektumfang und -management 4. Unzureichende Projektkommunikation 5. Fehlende Methodik

<p>After all this Why do projects fail?</p> <p>Rosenfeld, E. (n.d.)</p>	<ol style="list-style-type: none"> 1. Mangelhafte Benutzereingabe 2. Konflikte unter den Interessengruppen 3. Unklare Anforderungen 4. Schlechte Kosten- und Zeitschätzungen 5. Fähigkeiten, die nicht zum Aufgabenbereich passen 6. Versteckte Kosten von schlanker und sparsamer Vorgehensweise 7. Unzureichende Planung 8. Kommunikationsstörungen 9. Schlechte Software-Architektur 10. Späte Warnsignale für das Scheitern
<p>When BAD things happen to good projects</p> <p>Field, T. (1997)</p>	<ol style="list-style-type: none"> 1. Projektmanager verstehen die Bedürfnisse der Anwender nicht 2. Der Projektumfang ist schlecht definiert 3. Projektänderungen werden schlecht gehandhabt 4. Die gewählte Technologie ändert sich 5. Die Anforderungen ändern sich 6. Die Fristen sind unrealistisch gesetzt 7. Firmeninterner Projekt Widerstand 8. Dem Projekt fehlt es an Personen mit geeigneten Fähigkeiten 9. Das Projektmanagement missachtet erprobte Arbeitsmethoden zur Projektdurchführung

Unsere Analyse

8 Ursachen zeichnen sich als maßgeblich verantwortlich für das Scheitern von IT-Projekten ab. Wir fassen zusammen und erläutern anschließend die einzelnen Faktoren im Detail. Dabei schauen wir zuerst auf die Ursachen und machen anschließend Vorschläge, wie diese Fails vermeidbar werden:

1. Unzureichende oder fehlende Projektmanagement-Expertise
2. Fehlender frühzeitiger Einbezug von Endanwendern (Usern)
3. Unklarer oder unzureichend spezifizierter und definierter Projektumfang
4. Sich verändernde Projektziele und Anforderungen während der Projektumsetzung
5. Unrealistisch oder unzutreffende Aufwands- und Ressourcenschätzung zum Projektstart
6. Unzureichende Kommunikation und fehlende Zusammenarbeit im Team
7. Mangel an erforderlichem Team-Wissen und Fähigkeiten
8. Fehlende Unterstützung seitens der Führungsebene

Unzureichende oder fehlende Projektmanagement-Expertise

IT-Projekte können scheitern, wenn das Projektmanagement seiner Aufgabe unzureichend nachkommt. Was ist damit gemeint? Es liegt in der Aufgabe des Projektmanagements, dafür zu sorgen, dass Projektziele klar definiert sind, eine effektive Planung stattfindet, ein produktives Team aufgebaut wird, agile Vorgehensweisen etabliert werden, sowie dass ein kontinuierliches Risikomanagement stattfindet, um frühzeitige Anpassungen vorzunehmen, wenn das notwendig ist.

Das Projektmanagement schafft damit einerseits die Grundvoraussetzung, damit das Projekt überhaupt erfolgreich umgesetzt werden kann - andererseits überwacht es das Projekt auch hinsichtlich der Produktqualität und stellt sicher, dass Risiken nicht für unerwartete Probleme sorgen, indem es sie frühzeitig erkennt. Es ist aus diesem Grund vehement darauf zu achten, dass das Projektmanagement über die nötigen Eigenschaften und Fähigkeiten verfügt, die für

eine erfolgreiche Projektumsetzung erforderlich sind. Das sind bei IT-Projekten nicht ausschließlich die klassischen Projektmanagement-Fähigkeiten, sondern auch ein hohes technisches Verständnis.

Fehlender frühzeitiger Einbezug von Endanwendern

Viele Projekte werden entwickelt, ohne dass echte Endverbraucher einbezogen werden. Die kommen am Schluss dazu, so immer noch ein verbreiteter Fehlglaube. Der führt häufig dazu, dass theoretische Lösungsansätze entwickelt werden, die technisch Sinn ergeben, den tatsächlichen Bedürfnissen der Endverbraucher oder den realistischen Marktanforderungen überhaupt nicht entsprechen.

Wie kann dem entgegengewirkt werden? Ganz einfach: Schon während der Anforderungsanalyse sollen Endanwender einbezogen werden, die auch während der Projektumsetzung kontinuierlich Feedback geben und somit schnelle und wenig aufwändige Anpassungen möglich machen. So sinkt die Wahrscheinlichkeit rapide, dass ein Projekt umgesetzt wird, das am Markt versagt, weil es nicht das leistet, was die Anwender brauchen oder sich wünschen.

Unklarer oder unzureichend spezifizierter Projektumfang

Unklare oder unzureichend spezifizierte Anforderungen führen häufig zu einem langsamen Projektstart. Das Team muss immer wieder Zeit investieren, um die Anforderungen zu klären und zu verstehen, bevor es mit der Umsetzung beginnen kann. Oder, noch schlimmer: Das Team meint die Anforderungen zu verstehen, beginnt mit der Umsetzung, und hinterher stellt sich heraus, dass es am Kundenwunsch vorbeientwickelt hat. Das führt zu erheblichen Änderungsnotwendigkeiten während des Projektverlaufs. Keine Stunde, die dazu verwendet wird, die Projektanforderungen zu klären und sehr präzise zu definieren, ist verschwendete Zeit, im Gegenteil: sie hilft, später viel Zeit zu sparen und die reibungslose Umsetzung zu gewährleisten. Ein Workshop mit allen Beteiligten lohnt sich deshalb immer.

Zu Beginn des Projekts ist es dann entscheidend, einen ausreichenden Reifegrad für alle zu implementierenden Funktionalitäten und technischen Anforderungen festzulegen, das heißt: es reicht nicht, bestimmte Funktionalitäten erst dann im Detail zu spezifizieren, wenn man sie

konkret umzusetzen beginnt, sondern schon möglichst zu Projektstart. Das hilft nicht nur dabei, einzelne Projektschritte genauer zu verstehen, sondern auch, Überraschungen während der Umsetzung zu minimieren, die dann entstehen, wenn unterwegs klar wird, dass einzelne Umsetzungsschritte aufwendiger oder komplexer sind als ursprünglich vermutet.

In einer agilen Arbeitsweise erfolgt dann die weitergehende Detailspezifizierung von Themenblöcken iterativ. Dadurch kann das Team flexibel auf Änderungen reagieren, arbeitet jedoch dennoch mit klaren Anforderungen.

Wechselnde Projektziele und Umfang während der Projektumsetzung

Viele Projekte sind von einer schleichenden Ausweitung des Projektumfangs betroffen, wie auch von häufigen Änderungen der Anforderungen. Diese Projektvolatilität führt zu einer unzureichenden Vorhersehbarkeit und erzeugt damit eine Dynamik, die sich auf die Projekt-Ausführung und deren Ergebnisse auswirkt. Denn wenn ständig etwas geändert oder hinzugefügt werden soll, das anfangs nicht eingeplant war, braucht das Projekt länger, Ressourcen werden in Ergänzungen gebunden, die dann nicht mehr dem geplanten Projektfortschritt zur Verfügung stehen.

Aus diesem Grund ist es wichtig, bereits zu Projektstart einen ausreichenden Reifegrad aller betroffenen Funktionalitäten festzulegen, und Stakeholder sowie alle von fachlicher Seite involvierten Personen anzuweisen, betroffene Themen frühzeitig und ausführlich zu beleuchten, um spätere Richtungswechsel zu minimieren. Welche Schnittstellen müssen bedacht werden, was kann aus einem anderen Unternehmensbereich hinzukommen, der vielleicht nur marginal mit dem IT-Produkt arbeitet?

Ergeben sich dennoch erhebliche Änderungen, muss dies im Team frühzeitig kommuniziert und dokumentiert werden, und natürlich mit dem Kunden: Wenn dann gemeinsam entschieden wird, wie weitergearbeitet werden soll, lassen sich Schäden minimieren, und vielleicht wird mancher Änderungswunsch auf einen späteren Zeitpunkt verschoben, damit das Gesamtprojekt möglichst schnell zur Marktreife kommt.

Unrealistische oder unzutreffende Aufwands- und Ressourcenschätzung zum Projektstart

Werden der geschätzte Aufwand und die benötigten Ressourcen nicht präzise ermittelt, führt das regelmäßig zu Budgetüberschreitungen, ungeplanten Mehraufwänden, Verzögerungen im Zeitplan und Unzufriedenheit bei Stakeholdern, Kunden und im Team. Bei Verzögerungen in einem Projekt besteht zudem die Gefahr, dass die Umsetzungsqualität aufgrund des zeitlichen Drucks abnimmt. Natürlich gibt es dafür verschiedene Gründe: Einer mag sein, dass die Entwickler im Team und das Projektmanagement nicht über die ausreichende Erfahrung verfügen, um den tatsächlichen Aufwand einzuschätzen. Es soll auch vorkommen, dass Aufwände bewusst zu gering geschätzt werden, um unter mehreren Anbietern den Zuschlag zu bekommen. Für den Kunden sind beide Varianten ärgerlich, für das Entwickler-Team bedeuten sie Druck und Stress.

Diesem Risiko kann entgegengewirkt werden, indem Schätzungen auf einer sorgfältigen Analyse basieren müssen. Bei Bedarf kann es sich lohnen, erfahrene Experten hinzuzuziehen, die einschätzen können, ob der Aufwand auf einer realistischen Betrachtung beruht. Auch historische Daten aus früheren Projekten sollten berücksichtigt werden. Wir empfehlen außerdem, einen Puffer einzuplanen. IT-Projekte sind in der Regel komplex, und es können auch bei größter Sorgfalt unerwartete Herausforderungen auftreten.

Unzureichende Kommunikation und fehlende Zusammenarbeit im Team

Mangelnde Kommunikation kann zu Missverständnissen führen, sowohl hinsichtlich der Anforderungen als auch der Verantwortlichkeiten. Dies wiederum bewirkt Fehlinterpretationen und fehlerhafte Umsetzungen, ganz besonders, wenn unterschiedliche Erwartungen und Sichtweisen zwischen technischer und fachlicher Seite nicht frühzeitig identifiziert und geklärt werden. Außerdem können daraus entstehende persönliche Spannungen innerhalb der Teams die Effektivität der Arbeit stark beeinträchtigen.

Um dem vorzubeugen, sollten klare Kommunikationskanäle und -richtlinien implementiert werden. Regelmäßige gut strukturierte Teammeetings, in denen Fortschritte, Herausforderungen und zukünftige Pläne besprochen werden, fördern den Informationsfluss und ermöglichen die frühzeitige Identifizierung von Problemen. Kollaborationstools tragen zur Transparenz des Projektstatus bei und zentralisieren den Austausch von Ressourcen.

Durch die Festlegung klarer Rollen und Verantwortlichkeiten für jedes Teammitglied werden Unklarheiten bezüglich Zuständigkeiten minimiert.

Mangel an erforderlichlichem Team-Wissen und Fähigkeiten

Fehlendes Fachwissen und fehlende Fähigkeiten führen nicht selten dazu, dass das Team Schwierigkeiten hat, die Anforderungen des Projekts zu verstehen und effektiv umzusetzen. Das geschieht immer wieder, weil Technologien sich rapide entwickeln, und es dann möglicherweise nicht ausreichend Know-How im Team gibt, was neue Entwicklungen und ihre Einbindung angeht. Dies wiederum führt zu suboptimalen oder aufwendigen Lösungen, Qualitätsproblemen, Verzögerungen, oder sogar zum Scheitern des Projekts.

Vor der Projektumsetzung sollte daher eine gründliche Bedarfsanalyse durchgeführt werden, um sicherzustellen, dass alle notwendigen Kompetenzen vorhanden sind. Möglicherweise müssen zusätzliche Fachkräfte intern oder extern für das Projekt gewonnen oder Schulungen für bestehende Teammitglieder bereitgestellt werden. Zudem unterstützt ein kontinuierlicher Wissensaustausch in Form von regelmäßigen Meetings und Schulungen. Erfahrene Projektmitglieder können als Mentoren eingesetzt werden, um weniger erfahrenen Mitgliedern schnelle Hilfe bei Fragen oder Schwierigkeiten zu geben.

Fehlende Unterstützung seitens der Führungsebene

Hier geht es um die Prioritäten. Projekte, die nicht explizit klare Unterstützung von der oberen Führungsebene erhalten, werden häufig im ganzen Unternehmen nicht als prioritär betrachtet. Das führt oft dazu, dass sie vernachlässigt werden, ganz besonders dann, wenn das mittlere Management das Projekt nicht als wesentlich für das Unternehmen oder für seine eigene Leistungsbewertung ansieht und dementsprechend Ressourcen und Arbeitszeit auf Aktivitäten lenkt, die von der obersten Führungsebene eindeutig unterstützt werden. Ohne ausreichende Unterstützung seitens der Führungsebene können deshalb Budget- oder Personalengpässe im Projekt auftreten, nötige Informationen werden nicht rechtzeitig zur Verfügung gestellt, die Arbeit wird schwieriger.

Die Führungsebene sollte deshalb klar die Priorität des Projekts festlegen und anschließend allen Mitarbeitern im Unternehmen deutlich kommunizieren, dass das Projekt von

strategischer Bedeutung für das Unternehmen ist und unterstützt werden soll. Zudem ist es wichtig, alle relevanten Stakeholder aktiv in den Projektprozess einzubinden. Dies schafft von vorneherein Unterstützung und minimiert möglichen Widerstand. Die Führungsebene sollte sich regelmäßig über den Fortschritt des Projekts informieren und bei Bedarf , Anpassungen vornehmen - auch in der Kommunikation.

Fazit

Zusammenfassend lässt sich feststellen, dass gescheiterte Softwareentwicklungsprojekte auf eine Vielzahl von Ursachen zurückzuführen sind, die eng miteinander verknüpft sind. Das unzureichende Spezifikationsniveau zu Projektbeginn hat weitreichende Auswirkungen auf die gesamte Projektdurchführung, da es die Grundlage für das Verständnis der Anforderungen legt. Die Dynamik wechselnder Projektanforderungen während der Umsetzung bringt eine zusätzliche Schicht von Komplexität mit sich und erfordert eine flexible Anpassungsfähigkeit seitens des Teams.

Eine konsequente Anwendung von Projektmanagement-Fähigkeiten ist entscheidend, um den Projektablauf zu steuern, eine effiziente Umsetzung zu ermöglichen und etwaige Risiken frühzeitig zu erkennen - und zu beheben.

Unzureichende Kommunikation innerhalb des Teams beeinflusst maßgeblich die Zusammenarbeit und führt nicht selten zu Missverständnissen sowie Verzögerungen im Projektablauf. Diese Hürde wird verstärkt durch die fehlende Unterstützung seitens des Managements. Dieses sollte nicht nur die nötigen Ressourcen bereitstellen, sondern auch klare Prioritäten schaffen, damit alle Projektbeteiligten im Unternehmen Unterstützung und Informationen bekommen, wenn diese notwendig sind.

Der fehlende Einbezug von Endanwendern während des gesamten Prozesses kann zu einer Diskrepanz zwischen den erwarteten und tatsächlichen Ergebnissen führen und somit Nachbesserungen oder schlimmstenfalls Neuentwicklungen erfordern, wenn das Projekt scheinbar abgeschlossen ist.

Wir sind überzeugt: Um die Erfolgsaussichten von Softwareentwicklungsprojekten zu verbessern, bedarf es daher einer holistischen Herangehensweise: Wer nur auf die

Technologie schaut, übersieht mögliche Störfaktoren leicht. Kommunikative Faktoren, die strukturellen Voraussetzungen und die Skills der Projektmitglieder, über die rein technologischen Fähigkeiten hinaus (aber natürlich auch die Technologien betreffend) spielen eine gewichtige Rolle, wenn es um den Erfolg von IT-Projekten geht. Wer sie frühzeitig bedenkt, sich beim Spezifizieren der Projektanforderungen Zeit lässt und dem Projekt den Rahmen gibt, den es braucht, um erfolgreich umgesetzt zu werden, wird als Kunde viel Freude an seinem IT-Projekt haben.

GATE5, 2024.

A. Schrade

GATE5 GmbH, München

<https://gate5.digital>

Referenzen

Standish Group (2022), *The Chaos Report*. www.standishgroup.com.

Sauer, C., & Cuthbertson, C. (2003). *The State of IT Project Management in the UK*, Templeton College, Oxford.

Johnson, J., et al. (2001). *Collaborating on project success*. Retrieved from <http://www.softwaremag.com/>.

Fichter, D. (2003). *Why web projects fail*, [Online Journal], 27(4), 43. Retrieved Oct. 6, 2007 vom <http://www.ebscohost.com>.

Thibado, M. (2006). *Managing Successful Software Development Projects*. Ambient Consulting.

Al Neimat, Taimour. (2005) *Why it projects fail*

Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). *Identifying software project risks: An international Delphi study*. Journal of Management Information Systems, 17(4), 5–36.

Isfahani, K. (2010). *Why Projects Fail: Avoiding the Classic Pitfalls*. Oracle Corporation: Redwood Shores, CA

Rosenfeld, E. (n.d.). *After all this Why do projects fail?* Accessed March 12, 2024 vom <http://www.adaptivepartners.com/projfailb.htm>

Field, T. (1997). *When BAD things happen to good projects*, CIO, 55-62.