Dokumentation zur betrieblichen Projektarbeit

Projekttitel

Projektbetrieb

Autorin/Autor

Fachinformatikerin/Fachinformatiker für ...

Inhaltsverzeichnis

[1 Projektbeschreibung 5](#__RefHeading___Toc322_3725413944)

[2 Projekumfeld 5](#__RefHeading___Toc492_2387507712)

[3 Projektplanung 5](#__RefHeading___Toc169_3725413944)

[3.1 IST-Analyse 5](#__RefHeading___Toc494_2387507712)

[3.2 SOLL-Konzept 5](#__RefHeading___Toc496_2387507712)

[3.2.1 Produktqualität 5](#__RefHeading___Toc173_3725413944)

[3.2.1.1 Funktionalität 5](#__RefHeading___Toc175_3725413944)

[3.2.1.2 Leistungseffizienz 5](#__RefHeading___Toc177_3725413944)

[3.2.1.3 Kompatibilität 5](#__RefHeading___Toc179_3725413944)

[3.2.1.4 Benutzbarkeit 5](#__RefHeading___Toc181_3725413944)

[3.2.1.5 Zuverlässigkeit 5](#__RefHeading___Toc183_3725413944)

[3.2.1.6 Sicherheit 5](#__RefHeading___Toc185_3725413944)

[3.2.1.7 Wartbarkeit 5](#__RefHeading___Toc187_3725413944)

[3.2.1.8 Übertragbarkeit 6](#__RefHeading___Toc189_3725413944)

[3.2.2 Qualität im Einsatz 6](#__RefHeading___Toc191_3725413944)

[3.2.2.1 Effektivität 6](#__RefHeading___Toc193_3725413944)

[3.2.2.2 Effizienz 6](#__RefHeading___Toc498_2387507712)

[3.2.2.3 Anforderungen für Effizienz. 6](#__RefHeading___Toc500_2387507712)

[3.2.2.4 Zufriedenheit 6](#__RefHeading___Toc502_2387507712)

[3.2.2.5 Anforderungen für Zufriedenheit. 6](#__RefHeading___Toc504_2387507712)

[3.2.2.6 Freiheit von Risiken 6](#__RefHeading___Toc506_2387507712)

[3.2.2.7 Anforderungen für Freiheit von Risiken. 6](#__RefHeading___Toc508_2387507712)

[3.2.2.8 Erfüllung von Kontext-Anforderungen 6](#__RefHeading___Toc510_2387507712)

[3.2.2.9 Anforderungen für Erfüllung von Kontext-Anforderungen. 6](#__RefHeading___Toc512_2387507712)

[3.3 Ablaufplan 6](#__RefHeading___Toc514_2387507712)

[3.4 Ressourcenplanung 6](#__RefHeading___Toc516_2387507712)

[3.5 Kostenplanung 7](#__RefHeading___Toc518_2387507712)

[4 Projektdurchführung 7](#__RefHeading___Toc171_3725413944)

[4.1 Architekturdokumentation 7](#__RefHeading___Toc203_3725413944)

[4.2 Quellcode-Dokumentation 7](#__RefHeading___Toc205_3725413944)

[4.3 Testpläne und Testfälle 7](#__RefHeading___Toc207_3725413944)

[4.4 Nutzerdokumentation 7](#__RefHeading___Toc209_3725413944)

[4.5 Projektabschlussbericht 7](#__RefHeading___Toc211_3725413944)

[4.6 Änderungsprotokoll 7](#__RefHeading___Toc213_3725413944)

[4.7 Risikomanagementplan 7](#__RefHeading___Toc215_3725413944)

[4.8 Stakeholder-Kommunikationsplan 7](#__RefHeading___Toc217_3725413944)

[5 Projektauswertung 7](#__RefHeading___Toc520_2387507712)

[5.1 Zielerreichung 7](#__RefHeading___Toc522_2387507712)

[5.2 Soll-Ist-Vergleich 8](#__RefHeading___Toc524_2387507712)

[5.3 Optimierungs- und Erweiterungsmöglichkeiten 8](#__RefHeading___Toc526_2387507712)

[6 Quellenverzeichnis 8](#__RefHeading___Toc528_2387507712)

[7 Fußnoten 8](#__RefHeading___Toc530_2387507712)

[7.1 Abbildungen 8](#__RefHeading___Toc532_2387507712)

[8 Anhänge 8](#__RefHeading___Toc534_2387507712)

[A 8.1 Glossar 8](#__RefHeading___Toc536_2387507712)

[A 8.2 Pflichtenheft 8](#__RefHeading___Toc70_3725413944)

[A 8.3 PSP (Projektstrukturplan) 8](#__RefHeading___Toc72_3725413944)

[A 8.4 Gantt-Chart 8](#__RefHeading___Toc221_3725413944)

[A 8.5 Risikolog 9](#__RefHeading___Toc223_3725413944)

[A 8.6 Nutzwertanalyse 9](#__RefHeading___Toc225_3725413944)

# Projektbeschreibung

Die Projektbeschreibung sollte eine klare Übersicht über das Projekt bieten. Sie sollte den Umfang, die Hauptziele und die erwarteten Ergebnisse des Projekts umfassen. Es ist wichtig, sowohl kurzfristige als auch langfristige Ziele zu definieren, um den Projektverlauf besser steuern zu können.

Beispiel:

Angenommen, Sie leiten ein IT-Projekt zur Entwicklung einer mobilen Anwendung für ein Einzelhandelsgeschäft. Die kurzfristigen Ziele könnten die Entwicklung der Kernfunktionen und die Markteinführung innerhalb von sechs Monaten sein. Langfristige Ziele könnten die Erweiterung der App um zusätzliche Funktionen und die Erhöhung der Nutzerbasis um 25% im nächsten Jahr sein.

# Projektumfeld

Das Projektumfeld beschreibt die externen und internen Faktoren, die das Projekt beeinflussen könnten. Dazu gehören Stakeholder, Markttrends, technologische Herausforderungen und regulatorische Anforderungen. Ein gutes Verständnis des Projektumfelds hilft dabei, Risiken zu minimieren und Chancen zu maximieren.

Beispiel:

In dem oben genannten Projekt zur Entwicklung einer mobilen Anwendung könnte das Projektumfeld die Konkurrenz durch andere Einzelhandels-Apps, die Erwartungen der Stakeholder (z.B. Investoren, Endnutzer), technologische Beschränkungen und Datenschutzbestimmungen umfassen.

# Projektplanung

Der Projektplan ist das Herzstück der Projektmanagement-Dokumentation. Er legt den Rahmen für den gesamten Projektverlauf fest und sollte alle Phasen, Meilensteine und Ressourcen umfassen. Hier können auch Verweise auf Anhänge oder ein Pflichtenheft eingefügt werden, die weitere Details bieten.

Beispiel:

In einem Projekt zur Entwicklung einer Webanwendung könnte der Projektplan die Phasen der Konzeption, Entwicklung, Testung und Implementierung umfassen, jeweils mit Zeitrahmen und zugewiesenen Ressourcen.

## IST-Analyse

Die IST-Analyse ist eine Bestandsaufnahme des aktuellen Zustands. Sie sollte den aktuellen technologischen Stack, die vorhandenen Ressourcen und eventuelle Schwachstellen identifizieren.

Beispiel:  
In unserem Webanwendungsprojekt könnte die IST-Analyse zeigen, dass die aktuelle Website veraltet ist, nicht mobilfreundlich und langsame Ladezeiten hat.

## SOLL-Konzept

Das SOLL-Konzept listet die spezifischen Anforderungen des Projekts auf, die in Übereinstimmung mit ISO 25010 ausgewählt werden sollten. Dazu können Funktionalität, Zuverlässigkeit, Benutzerfreundlichkeit usw. gehören. Ein Verweis auf ein Pflichtenheft kann hier nützlich sein, um detaillierte Anforderungen zu klären.

Beispiel:  
Für unser Webanwendungsprojekt könnten die Anforderungen nach ISO 25010 die Entwicklung einer benutzerfreundlichen Oberfläche, die Implementierung von Sicherheitsprotokollen und die Gewährleistung einer hohen Verfügbarkeit sein.

### Produktqualität

### Funktionalität

Die Funktionalität bezieht sich auf die spezifischen Features und Fähigkeiten des Produkts. Hier sollten alle erforderlichen Funktionen, die das Produkt erfüllen muss, aufgelistet werden.

Beispiel:  
In einer E-Commerce-App könnten die Anforderungen für die Funktionalität die Implementierung eines Warenkorbs, einer Suchfunktion und verschiedener Zahlungsoptionen sein.

### Leistungseffizienz

Dieser Abschnitt sollte die Anforderungen an die Performance des Systems festlegen, wie z.B. Antwortzeiten oder Ressourcenverbrauch.

Beispiel:  
Die E-Commerce-App sollte in der Lage sein, mindestens 1000 gleichzeitige Nutzer zu unterstützen, ohne dass die Antwortzeit 2 Sekunden überschreitet.

### Kompatibilität

Hier werden die Anforderungen für die Interoperabilität des Produkts mit anderen Systemen oder Softwareversionen festgelegt.

Beispiel:  
Die App sollte sowohl auf Android als auch auf iOS lauffähig sein und mit den gängigen Webbrowsern kompatibel sein.

### Benutzbarkeit

Dieser Abschnitt legt die Anforderungen an die Benutzerfreundlichkeit des Produkts fest.

Beispiel:  
Die App sollte eine intuitive Benutzeroberfläche haben und die wichtigsten Funktionen sollten innerhalb von drei Klicks erreichbar sein.

### Zuverlässigkeit

Hier werden die Anforderungen an die Zuverlässigkeit des Systems festgelegt, wie z.B. Verfügbarkeit und Fehlertoleranz.

Beispiel:  
Die App sollte eine Verfügbarkeit von 99,9% haben und in der Lage sein, sich von kleineren Fehlern selbstständig zu erholen.

### Sicherheit

Dieser Abschnitt sollte die Sicherheitsanforderungen des Produkts, wie z.B. Datenverschlüsselung und Authentifizierung, festlegen.

Beispiel:

Die App sollte eine Zwei-Faktor-Authentifizierung unterstützen und alle sensiblen Daten müssen verschlüsselt sein.

### Wartbarkeit

Hier werden die Anforderungen an die Wartbarkeit des Systems festgelegt, wie z.B. Modularität und Dokumentation.

Beispiel:  
Der Code der App sollte gut dokumentiert sein und in modularer Form geschrieben sein, um die Wartung zu erleichtern.

### Übertragbarkeit

Dieser Abschnitt legt die Anforderungen an die Portabilität des Produkts fest, also wie einfach es ist, das Produkt auf verschiedenen Systemen oder in verschiedenen Umgebungen zu verwenden.

Beispiel:  
Die App sollte so entwickelt sein, dass sie leicht auf andere Märkte oder Plattformen übertragen werden kann, ohne große Änderungen am Code vornehmen zu müssen.

### Qualität im Einsatz

### Effektivität

Effektivität bezieht sich darauf, wie gut das Produkt seine beabsichtigten Ziele erreicht. Hier sollten die Kriterien festgelegt werden, anhand derer die Effektivität gemessen wird.

Beispiel:  
In einer E-Learning-Plattform könnte die Effektivität durch die Abschlussrate der Kurse und die Verbesserung der Fähigkeiten der Benutzer gemessen werden.

### Effizienz

Effizienz betrachtet, wie gut das Produkt seine Ziele mit minimalem Ressourcenaufwand erreicht.

Beispiel:  
Die E-Learning-Plattform sollte in der Lage sein, Videos schnell zu laden und gleichzeitig den Bandbreitenverbrauch zu minimieren.

### Zufriedenheit

Zufriedenheit bezieht sich auf das Benutzererlebnis und wie zufrieden die Benutzer mit dem Produkt sind.

Beispiel:  
Die Plattform sollte eine benutzerfreundliche Oberfläche und schnellen Kundensupport bieten, um die Benutzerzufriedenheit zu erhöhen.

### Freiheit von Risiken

Dieser Abschnitt sollte die Anforderungen für die Minimierung von Risiken, wie z.B. Datenschutzverletzungen oder Systemausfälle, festlegen.

Beispiel:  
Die Plattform sollte regelmäßige Sicherheitsaudits durchführen und einen Notfallwiederherstellungsplan haben.

### Erfüllung von Kontext-Anforderungen

Dieser Abschnitt sollte die Anforderungen für die Anpassungsfähigkeit des Produkts an verschiedene Kontexte oder Einsatzszenarien festlegen.

Beispiel:  
Die E-Learning-Plattform sollte in der Lage sein, sowohl in einem Unternehmenskontext als auch für individuelle Lernende effektiv zu funktionieren.

## Ablaufplan

Der Ablaufplan ist in einem Ein-Personen-Projekt besonders wichtig, da er eine klare Struktur und Orientierung bietet. Er stellt die zeitliche Abfolge der einzelnen Aktivitäten dar und ist eng mit dem Projektzeitplan und dem Gantt-Chart verknüpft. In diesem Abschnitt werden spezifische Details zu den Aktivitäten, deren Abhängigkeiten und dem geplanten Zeitrahmen erläutert.

Beispiel:  
Angenommen, das Projektziel ist die Erstellung einer einfachen Website. Der Ablaufplan könnte folgendermaßen aussehen:

* Phase 1: Konzeption (Woche 1)
  + Meilenstein: Fertigstellung des Konzepts
* Phase 2: Design (Woche 2)
  + Meilenstein: Fertigstellung des Designs
* Phase 3: Entwicklung (Woche 3-4)
  + Meilenstein: Fertigstellung der ersten Version
* Phase 4: Testen (Woche 5)
  + Meilenstein: Abschluss der Testphase
* Phase 5: Implementierung (Woche 6)
  + Meilenstein: Website geht live

Jede Phase hat spezifische Aktivitäten und Abhängigkeiten. Zum Beispiel ist die Entwicklung abhängig von der Fertigstellung des Designs, und das Testen kann erst beginnen, wenn die Entwicklung abgeschlossen ist.

## Ressourcenplanung

In einem Ein-Personen-Projekt ist die effektive Zuweisung von Zeit und Fähigkeiten entscheidend. Materielle Ressourcen wie Hardware und Software sind weiterhin wichtig, aber die Planung muss auch die begrenzte Verfügbarkeit der einzigen Person berücksichtigen, die am Projekt arbeitet.

Beispiel:  
In einem Ein-Personen-Projekt zur Erstellung einer Website könnte die benötigte Hardware ein Laptop und die benötigte Software ein Texteditor und ein Webserver sein. Die personelle Ressource ist der Einzelunternehmer selbst, der alle Aufgaben, von der Konzeption bis zur Implementierung, übernimmt.

## Kostenplanung

Die Kostenplanung in einem Ein-Personen-Projekt ist oft einfacher, da weniger externe Ressourcen und Mitarbeiter involviert sind. Dennoch ist eine detaillierte Aufschlüsselung der voraussichtlichen Projektkosten wichtig, um das Budget effektiv zu verwalten.

Beispiel:  
Die Fixkosten könnten die Anschaffung von Softwarelizenzen und die Einrichtung eines Webservers umfassen. Variable Kosten könnten sich aus dem Bedarf an zusätzlichen Software-Tools oder Dienstleistungen ergeben, die während des Projektverlaufs benötigt werden. Da nur eine Person am Projekt arbeitet, könnten die Arbeitskosten als "Opportunitätskosten" betrachtet werden, also als entgangene Einnahmen aus anderen Projekten oder Tätigkeiten.

# Projektdurchführung

## Architekturdokumentation

Dieser Abschnitt sollte die technische Architektur des Projekts detailliert beschreiben, einschließlich der verwendeten Technologien und der Beziehungen zwischen verschiedenen Komponenten.

Beispiel:  
In einem Projekt zur Entwicklung einer Webanwendung könnte die Architektur eine Frontend-Komponente in React und eine Backend-Komponente in Node.js umfassen.

## Quellcode-Dokumentation

Der Quellcode sollte gut kommentiert sein, und dieser Abschnitt sollte zusätzliche Erklärungen und Kontext bieten.

Beispiel:  
Kommentare im Code könnten die Funktion jedes Moduls und die Beziehungen zwischen ihnen erläutern.

## Testpläne und Testfälle

Dieser Abschnitt sollte die Teststrategie, die verschiedenen Testfälle und die erwarteten Ergebnisse beschreiben.

Beispiel:  
Ein Testplan könnte Unit-Tests für einzelne Funktionen und Integrationstests für das gesamte System umfassen.

## Nutzerdokumentation

Dieser Abschnitt sollte eine klare Anleitung für den Endbenutzer bieten, wie die Software zu verwenden ist.

Beispiel:  
Eine schrittweise Anleitung könnte dem Benutzer zeigen, wie er ein Konto erstellt und Funktionen der Anwendung nutzt.

## Änderungsprotokoll

Alle Änderungen am Projekt sollten in diesem Abschnitt dokumentiert werden.

Beispiel:  
Jede neue Version der Software könnte hier mit den jeweiligen Änderungen und Fixes aufgelistet werden.

## Risikomanagementplan

Dieser Abschnitt sollte alle potenziellen Risiken identifizieren und Pläne für ihre Minderung skizzieren. Hier kann auch auf das Risikolog (s. S. 11) Bezug genommen werden.

Beispiel:  
Risiken könnten Verzögerungen im Zeitplan oder technische Schwierigkeiten sein, und der Plan könnte Notfallmaßnahmen enthalten.

## Stakeholder-Kommunikationsplan

In einem Ein-Personen-Projekt könnte dieser Abschnitt weniger relevant sein, es sei denn, es gibt externe Stakeholder. In diesem Fall sollte der Kommunikationsplan festlegen, wie und wann mit ihnen kommuniziert wird.

Beispiel:  
Wenn es externe Investoren oder Kunden gibt, könnte der Plan regelmäßige Updates und Meetings vorsehen.

# Projektauswertung

Dieser Abschnitt dient der umfassenden Analyse und Bewertung der Projektleistung und -ergebnisse.

## Zielerreichung

In diesem Abschnitt wird bewertet, inwiefern die ursprünglichen Projektziele erreicht wurden. Dies kann quantitativ oder qualitativ erfolgen, je nach den spezifischen Zielen und KPIs des Projekts.

Beispiel:  
Wenn das Ziel die Entwicklung einer Webanwendung mit mindestens 100 aktiven Benutzern war, könnte die Zielerreichung quantitativ durch die tatsächliche Nutzerzahl bewertet werden.

## Soll-Ist-Vergleich

Dieser Unterabschnitt sollte eine detaillierte Analyse der Abweichungen zwischen den geplanten und den tatsächlich erreichten Zielen bieten.

Beispiel:  
Wenn die geplante Entwicklungszeit 6 Wochen war, aber tatsächlich 8 Wochen benötigt wurden, sollte der Grund für diese Abweichung analysiert werden.

## Optimierungs- und Erweiterungsmöglichkeiten

Dieser Abschnitt sollte potenzielle Bereiche für Verbesserungen und mögliche Erweiterungen des Projekts oder des entwickelten Produkts auflisten.

Beispiel:  
Mögliche Optimierungen könnten die Verbesserung der Benutzeroberfläche oder die Implementierung zusätzlicher Funktionen sein. Erweiterungsmöglichkeiten könnten die Entwicklung einer mobilen App oder die Internationalisierung der Webanwendung sein.

# Quellenverzeichnis

## Fußnoten

Dieser Abschnitt sollte alle Fußnoten auflisten, die im Dokument verwendet wurden, um zusätzliche Erläuterungen oder Bezüge zu den jeweiligen Quellen zu bieten.

Beispiel:  
Siehe Smith, J. (2020) für Best Practices in der Webentwicklung.  
Basierend auf den offiziellen React-Dokumenten.

## Abbildungen

Dieser Unterabschnitt sollte alle im Dokument verwendeten Abbildungen, Diagramme und Tabellen auflisten, inklusive einer kurzen Beschreibung und der Quellenangabe, falls zutreffend.

Beispiel:  
Abbildung 1: Architekturdiagramm der Webanwendung. Quelle: Eigene Darstellung.  
Tabelle 1: Vergleich von geplanten und tatsächlichen Entwicklungszeiten. Quelle: Eigene Erhebung.

# Anhänge

* 1. Glossar

Hier werden wichtige Begriffe definiert.

* 1. Pflichtenheft

Dieser Anhang enthält das Pflichtenheft

* 1. PSP (Projektstrukturplan)

Dieser Anhang enthält den Projektstrukturplan.

* 1. Gantt-Chart

Dieser Anhang enthält das Gantt-Chart für das Projekt.

* 1. Risikolog

Dieser Anhang enthält das Risikolog für das Projekt.

* 1. Nutzwertanalyse

Dieser Anhang enthält die Nutzwertanalyse für das Projekt.