

# Erstellen eines Lastenhefts

## Inhalt

Einleitung .....	1
Zielsetzung, Stakeholder und Zeitplan .....	2
Funktionale und nicht-funktionale Anforderungen .....	4
Funktionale Anforderungen .....	5
User Story und Use Cases.....	5
Wie kommt man zu funktionalen Anforderungen? .....	6
Zusammenfassung .....	7
Risikomanagement.....	8
Wie wird Risikomanagement betrieben? .....	9
Beispiel Risiko in einem Risikoregister .....	9
Zusammenfassung .....	9

## Einleitung

In der heutigen Zeit ist die Nutzung von digitalen Tools im Hotelmanagement unverzichtbar geworden. Die Einführung einer Hotelsoftware kann jedoch eine komplexe und anspruchsvolle Aufgabe darstellen, die nur mit einer detaillierten Planung und einem umfassenden Verständnis der Anforderungen und Stakeholder erfolgreich umgesetzt werden kann.

Die Zielsetzung bei der Einführung einer Hotelsoftware ist es, eine effiziente und benutzerfreundliche Lösung zu schaffen, die den spezifischen Bedürfnissen des Hotels entspricht. Um dieses Ziel zu erreichen, müssen die Anforderungen der Stakeholder sorgfältig ermittelt werden. Dazu gehören sowohl interne Mitarbeiter wie auch externe Partner und Gäste.

Verantwortlichkeiten der Stakeholder müssen klar definiert werden, um sicherzustellen, dass alle notwendigen Aufgaben und Entscheidungen zeitnah und effektiv umgesetzt werden.

Das Lastenheft ist ein wesentlicher Bestandteil des Projekts. Es definiert die funktionale und nicht-funktionale Anforderungen an die Software sowie die vertraglichen Vereinbarungen zwischen dem Hotel und dem Anbieter. Eine klare Definition der Anforderungen und Vereinbarungen ist entscheidend für den Erfolg des Projekts.

Ein weiterer wichtiger Aspekt bei der Einführung einer Hotelsoftware ist das Risikomanagement. Hierbei geht es darum, potenzielle Risiken und deren Auswirkungen zu identifizieren und geeignete Maßnahmen zu ergreifen, um diese Risiken zu minimieren oder zu eliminieren. Dabei sollten nicht nur technische Risiken, sondern auch die möglichen Auswirkungen auf Mitarbeiter und Gäste berücksichtigt werden.

Insgesamt ist es wichtig, dass alle Aspekte des Projekts sorgfältig geplant und koordiniert werden, um sicherzustellen, dass die neue Hotelsoftware effektiv und erfolgreich implementiert wird.

## Zielsetzung, Stakeholder und Zeitplan

Eine detaillierte Zielsetzung ist wichtig, um sicherzustellen, dass alle Beteiligten ein gemeinsames Verständnis darüber haben, was erreicht werden soll und wie der Erfolg des Projekts gemessen wird. Eine klare Zielsetzung hilft, das Projekt auf Kurs zu halten, Entscheidungen zu treffen und Ressourcen effektiv zu nutzen.

Ein Beispiel für eine Zielsetzung im Zusammenhang mit der Einführung einer Hotelsoftware könnte wie folgt lauten:

"Das Ziel dieses Projekts ist es, eine neue Hotelsoftware erfolgreich zu implementieren, um die Effizienz und Qualität der Hotelprozesse zu steigern und somit eine höhere Gästezufriedenheit zu erreichen. Die neue Hotelsoftware soll alle wichtigen Funktionen und Anforderungen des Hotels erfüllen und eine benutzerfreundliche Oberfläche bieten. Dabei soll besonders auf eine einfache Bedienbarkeit für das Hotelpersonal geachtet werden. Die Einführung der neuen Software soll ohne Beeinträchtigung des täglichen Hotelbetriebs erfolgen und innerhalb des vorgegebenen Budgets und Zeitrahmens abgeschlossen werden. Der Erfolg des Projekts wird anhand von messbaren Kriterien, wie der Akzeptanz durch die Mitarbeiter, der Verbesserung der Gästezufriedenheit und der Steigerung der Effizienz und Produktivität gemessen werden."

Um die in der Zielsetzung definierten Metriken für den Erfolg der Hotelsoftware-Einführung messen zu können, müssen zunächst klare Kriterien und Indikatoren für jede Metrik definiert werden. Diese Kriterien können zum Beispiel sein:

- Gästezufriedenheit: Bewertungen und Feedback der Gäste, Anzahl der Beschwerden, wiederkehrende Gäste
- Effizienz und Produktivität: Zeitersparnis bei der Abwicklung von Aufgaben, Anzahl der Buchungen pro Tag/Stunde, Auslastung des Hotels
- Akzeptanz durch Mitarbeiter: Feedback der Mitarbeiter, Schulungsbedarf, Anzahl der Mitarbeiter, die die Software aktiv nutzen
- Datenqualität: Die Qualität der Daten, die von der Hotelsoftware erfasst und verarbeitet werden, ist ein wichtiger Erfolgsfaktor. Hohe Datenqualität sorgt für genaue und verlässliche Informationen, die bei Entscheidungen und Planungen unterstützen.
- Sicherheit: Eine Hotelsoftware muss sicher und zuverlässig sein, um den Schutz der Daten und die Integrität der Systeme zu gewährleisten. Ein wichtiger Erfolgsfaktor ist die Vermeidung von Datenlecks und Systemausfällen.
- Wartung und Support: Eine Hotelsoftware erfordert kontinuierliche Wartung und Unterstützung, um sicherzustellen, dass sie reibungslos funktioniert und alle Anforderungen erfüllt werden. Ein guter Kundensupport und regelmäßige Updates sind daher entscheidende Faktoren für den Erfolg.

Um diese Kriterien zu messen, können verschiedene Methoden eingesetzt werden, wie zum Beispiel Umfragen bei Gästen und Mitarbeitern, Analyse von Buchungs- und Abrechnungsdaten, Beobachtung der Arbeitsprozesse und Schulungsbedarf.

Wichtig ist dabei, dass die Messung kontinuierlich stattfindet und die Ergebnisse regelmäßig ausgewertet werden, um gegebenenfalls Anpassungen vornehmen zu können. Es ist auch empfehlenswert, bereits vor der Einführung klare Ziele und Erwartungen zu definieren und diese gemeinsam mit allen beteiligten Personen zu kommunizieren.

Um den Erfolg eines Projektes sicherzustellen, ist es nicht nur wichtig, klare Metriken zur Messung des Fortschritts und des Ergebnisses zu definieren, sondern auch die Verantwortlichkeiten der Stakeholder im Projekt zu definieren. Jeder Stakeholder sollte seine Rolle und seine Verantwortlichkeiten kennen, um sicherzustellen, dass das Projekt in die gewünschte Richtung läuft. Wenn alle Stakeholder wissen, was von ihnen erwartet wird und welche Verantwortlichkeiten sie haben, kann das Projekt reibungsloser und erfolgreicher verlaufen. Daher ist es unerlässlich, die Verantwortlichkeiten im Lastenheft festzuhalten und die Stakeholder klar zu definieren.

Um die Verantwortlichkeiten im Lastenheft festzuhalten, können Sie eine Tabelle erstellen, in der Sie die verschiedenen Anforderungen und die dazugehörigen Verantwortlichkeiten aufzeichnen. Hier sind einige Beispiele, wie Sie Stakeholder in Ihrem Lastenheft definieren können:

- Projektleiter: Der Projektleiter ist verantwortlich für die Koordination des Projekts, die Festlegung der Ziele und die Überwachung des Fortschritts.
- IT-Manager: Der IT-Manager ist verantwortlich für die technische Umsetzung der Anforderungen und die Integration der neuen Hotelsoftware in die bestehende IT-Infrastruktur.
- Finanzabteilung: Die Finanzabteilung ist verantwortlich für die Budgetierung und Überwachung der Projektkosten sowie für die Bewertung des ROI (Return on Investment) des neuen Systems.
- Rezeptionsleiter: Der Rezeptionsleiter ist verantwortlich für die Schulung der Mitarbeiter und die Überwachung der Nutzung der neuen Hotelsoftware.
- Gäste: Gäste sind ebenfalls Stakeholder, da das System auch auf deren Bedürfnisse ausgelegt sein sollte. Sie können beispielsweise Feedback geben und durch die Nutzung der Software die Effektivität des Systems beeinflussen.

Ein weiterer wichtiger Faktor für den Erfolg der Einführung einer Hotelsoftware ist die Einhaltung des Zeitplans. Ein klar definierter Zeitplan kann sicherstellen, dass das Projekt pünktlich und innerhalb des Budgets abgeschlossen wird. Es ist wichtig, dass jeder Stakeholder seinen Beitrag leistet, um den Zeitplan einzuhalten. Die Stakeholder sollten sich regelmäßig treffen, um den Fortschritt des Projekts zu besprechen und sicherzustellen, dass es in der vorgegebenen Zeit abgeschlossen wird. Wenn Probleme auftreten, die den Zeitplan beeinträchtigen, müssen sie umgehend identifiziert und gelöst werden. Eine klare Definition der Verantwortlichkeiten im Lastenheft kann dazu beitragen, dass jeder Stakeholder seine Rolle im Projekt versteht und sicherstellt, dass alle Aufgaben pünktlich und in der erforderlichen Qualität ausgeführt werden.

Zielsetzung, Stakeholder und Zeitplan sind entscheidende Aspekte bei der Einführung einer Hotelsoftware. Eine klare Zielsetzung gibt den Rahmen für das Projekt vor und ermöglicht eine klare Definition der Anforderungen. Die Stakeholder sollten von Anfang an definiert und in das Projekt involviert werden, um sicherzustellen, dass ihre Anforderungen berücksichtigt werden und eine erfolgreiche Implementierung erreicht werden kann. Ein detaillierter Zeitplan hilft dabei, das Projekt im Zeitrahmen zu halten und sicherzustellen, dass wichtige Meilensteine erreicht werden. Die Verantwortlichkeiten der Stakeholder sollten im Zeitplan deutlich festgehalten werden, um sicherzustellen, dass jeder weiß, welche Aufgaben zu erledigen und welche Fristen einzuhalten sind. Insgesamt sind Zielsetzung, Stakeholder und Zeitplan entscheidende Faktoren, um sicherzustellen, dass die Einführung einer Hotelsoftware erfolgreich verläuft.

## Funktionale und nicht-funktionale Anforderungen

Funktionale und nicht-funktionale Anforderungen sind zwei Arten von Anforderungen, die bei der Entwicklung bzw. Einführung von Software oder Systemen eine wichtige Rolle spielen. Im Folgenden werden die Unterschiede zwischen funktionalen und nicht-funktionalen Anforderungen erläutert.

Funktionale Anforderungen beschreiben die Funktionalität oder die Aufgaben, die das System oder die Software ausführen soll. Sie beschreiben, welche Funktionen das System haben soll, welche Aktionen es ausführen soll und wie es auf bestimmte Benutzereingaben reagieren soll. Funktionale Anforderungen sind somit darauf ausgerichtet, was das System tun soll, und stellen eine Art Vertrag zwischen dem Kunden und dem Software-Unternehmen dar.

Beispiele für funktionale Anforderungen im Kontext einer Hotelsoftware sind:

- Zimmerbuchung: Die Software muss in der Lage sein, Zimmerbuchungen für Gäste entgegenzunehmen, zu verwalten und zu stornieren.
- Check-in / Check-out: Die Software sollte einen Check-in- und Check-out-Prozess für Gäste bieten, um ihre Ankunft und Abreise zu verfolgen.

Funktionale Anforderungen können diverse Abstufungen des Details Grads aufweisen:

### **Miteinbeziehen von Akteuren:**

- Zimmerbuchung: Diese Anforderung stellt sicher, dass die Software in der Lage ist, Zimmerbuchungen für Gäste entgegenzunehmen, zu verwalten und zu stornieren. Die Funktion erfordert eine Benutzeroberfläche, die es dem Front-Office-Personal ermöglicht, Buchungen für Gäste zu erstellen, zu bearbeiten oder zu stornieren, sowie eine Datenbank, die Informationen über Zimmerverfügbarkeit, Preise und Reservierungen speichert.
- Check-in / Check-out: Diese Anforderung stellt sicher, dass die Software einen reibungslosen Check-in- und Check-out-Prozess für Gäste bietet, um ihre Ankunft und Abreise zu verfolgen. Die Funktion erfordert eine Benutzeroberfläche, die es dem Front-Office-Personal ermöglicht, Gästeeinformationen aufzunehmen und zu aktualisieren sowie eine Schnittstelle zur Zimmerverwaltung, um den Status von Zimmern und den Zeitpunkt ihrer Verfügbarkeit zu verwalten.

### **Miteinbeziehen von Prozessschritten:**

- Check-in / Check-out: Diese Anforderung stellt sicher, dass die Software einen reibungslosen Check-in- und Check-out-Prozess für Gäste bietet, um ihre Ankunft und Abreise zu verfolgen. Die Funktion erfordert eine Benutzeroberfläche, die es dem Front-Office-Personal ermöglicht, Gästeeinformationen aufzunehmen und zu aktualisieren sowie eine Schnittstelle zur Zimmerverwaltung, um den Status von Zimmern und den Zeitpunkt ihrer Verfügbarkeit zu verwalten.
  1. Der Gast meldet sich an der Rezeption und gibt seine Buchungs- oder Identifikationsdaten an.
  2. Das Front-Office-Personal prüft die Buchungsinformationen und aktualisiert den Status des Zimmers auf "belegt".
  3. Das Front-Office-Personal händigt dem Gast die Zimmerkarte aus und gibt ihm Informationen zu den Einrichtungen und Services des Hotels.
  4. Das Front-Office-Personal stellt dem Gast eine Rechnung für den Aufenthalt aus und nimmt die Zahlung entgegen.

Wie man funktionale Anforderungen ausarbeiten kann und welche Form diese Anforderungen aufweisen können, wird nachfolgend genauer erläutert.

Nicht-funktionale Anforderungen hingegen beschreiben Eigenschaften oder Merkmale des Systems oder der Software, die nicht unmittelbar mit seiner Funktionalität oder den ausgeführten Aufgaben zusammenhängen. Sie beschreiben die Art und Weise, wie das System seine Funktionen ausführt und wie es sich verhält. Nicht-funktionale Anforderungen können Einfluss auf die Qualität des Systems und die Benutzererfahrung haben.

Beispiele für nicht-funktionale Anforderungen im Kontext einer Hotelsoftware sind:

- Die Software sollte schnell arbeiten und kurze Antwortzeiten haben.
- Die Benutzeroberfläche sollte einfach zu bedienen sein.
- Das System sollte sicher und zuverlässig sein.

Im Gegensatz zu funktionalen Anforderungen sind nicht-funktionale Anforderungen schwerer zu quantifizieren und zu messen, da sie sich auf Qualität, Benutzererfahrung und andere Eigenschaften beziehen, die nicht direkt messbar sind. Sie sind jedoch genauso wichtig wie funktionale Anforderungen, da sie dazu beitragen, dass das System insgesamt effektiv und benutzerfreundlich ist.

Zusammenfassend lässt sich sagen, dass funktionale Anforderungen beschreiben, was das System tun soll, während nicht-funktionale Anforderungen beschreiben, wie es seine Aufgaben ausführt und welche Eigenschaften es hat. Beide Arten von Anforderungen sind wichtig für die erfolgreiche Entwicklung bzw. Einführung eines Systems oder einer Software.

## Funktionale Anforderungen

Funktionale Anforderungen beschreiben, welche Funktionen ein System oder eine Software ausführen soll. Sie legen fest, welche Aufgaben das System erledigen muss und wie es auf bestimmte Benutzereingaben reagieren soll. Funktionale Anforderungen sind normalerweise in Form von User Stories oder Use Cases formuliert, um den Fokus auf die Bedürfnisse der Benutzer zu legen. Eine typische User Story oder ein Use Case enthält eine Beschreibung der Funktion, die Anforderungen an die Eingabe und eine Beschreibung der erwarteten Ausgabe.

### User Story und Use Cases

Eine User Story ist eine kurze, einfache Beschreibung der Funktionalität eines Systems aus der Perspektive eines Benutzers. Sie beschreibt, welche Aufgabe der Benutzer ausführen möchte, warum er diese Aufgabe ausführen möchte und welche Vorteile er sich von der Ausführung dieser Aufgabe erhofft. User Stories sind oft in einer speziellen Schreibweise formuliert, wie zum Beispiel "Als <Benutzer> möchte ich <Funktionalität>, damit ich <Ziel erreiche>". Ein Beispiel für eine User Story in einem Hotelsoftware-Kontext könnte lauten: "Als Receptionist möchte ich Gäste suchen und Buchungen vornehmen können, damit ich den Gästen bei der Buchung helfen kann und Buchungen effizient verwalten kann."

Ein Use Case beschreibt die Interaktion zwischen dem Benutzer und dem System und wie das System auf Benutzereingaben reagieren sollte. Ein Use Case beschreibt normalerweise einen Ablauf, wie eine bestimmte Aufgabe ausgeführt wird, von Anfang bis Ende. Er gibt detailliert an, welche Schritte ausgeführt werden müssen, um das Ziel zu erreichen. Ein Beispiel für einen Use Case in einem Hotelsoftware-Kontext wäre ein Check-in-Use-Case, der den Ablauf beschreibt, wenn ein Gast eincheckt, einschließlich Schritte wie die Identifizierung des Gastes, die Überprüfung der Verfügbarkeit des Zimmers und die Zuweisung des Zimmers.

*Beispiel Check-in-Use-Case:*

Name des Use Cases: Check-in

Ziel: Ein Gast soll erfolgreich eingeecheckt werden.

Akteure: Rezeptionist, Gast

Vorbedingungen: Der Gast ist im System registriert und hat eine Buchung.

Nachbedingungen: Der Gast ist eingeecheckt, das Zimmer wurde ihm zugewiesen.

Beschreibung des Ablaufs:

1. Der Rezeptionist begrüßt den Gast und fragt nach seinem Namen und seiner Buchungsnummer.
2. Der Rezeptionist sucht nach dem Gast im System und bestätigt, dass er eine Buchung hat.
3. Der Rezeptionist fragt den Gast nach seiner ID oder einem gültigen Ausweisdokument und prüft die Identität des Gastes.
4. Der Rezeptionist prüft die Verfügbarkeit des Zimmers, das dem Gast zugewiesen werden soll.
5. Wenn das Zimmer verfügbar ist, teilt der Rezeptionist dem Gast die Zimmerdetails (z. B. Zimmer-Nummer, Etage, Zugangskarte) mit und bittet ihn, das Zimmer zu überprüfen.
6. Der Rezeptionist teilt dem Gast die Check-out-Zeit mit und bittet ihn, das Melde-Formular zu unterschreiben.
7. Der Rezeptionist erstellt ein Check-in im System und weist dem Gast das Zimmer zu.
8. Der Gast erhält die Schlüsselkarte oder den Zugangscode für das Zimmer.
9. Der Rezeptionist informiert den Gast über die Annehmlichkeiten und Services des Hotels und beantwortet eventuelle Fragen des Gastes.

Der Hauptunterschied zwischen einer User Story und einem Use Case liegt in der Detailtiefe und dem Schwerpunkt der Beschreibung. Während User Stories die Bedürfnisse der Benutzer hervorheben und eine kurze, prägnante Beschreibung der Funktionalität bieten, legen Use Cases einen Fokus auf den Interaktionsablauf zwischen Benutzer und System und beschreiben detailliert die notwendigen Schritte, um das Ziel zu erreichen. Beide Beschreibungen können nützlich sein, um die Anforderungen aus der Perspektive des Benutzers zu beschreiben, und es kann sinnvoll sein, beide Arten von Beschreibungen zu verwenden, um sicherzustellen, dass die Anforderungen vollständig und verständlich sind.

### Wie kommt man zu funktionalen Anforderungen?

Als Kunde kann man zu funktionalen Anforderungen auf verschiedene Weise kommen. Eine Möglichkeit ist, eine Anforderungsanalyse durchzuführen, indem man die Geschäftsprozesse analysiert und identifiziert, welche Funktionen das System ausführen muss, um diese Prozesse zu unterstützen. Eine andere Möglichkeit ist, Workshops mit den Benutzern oder Stakeholdern durchzuführen, um ihre Anforderungen und Bedürfnisse zu ermitteln.

Es gibt auch verschiedene Techniken, die bei der Identifikation funktionaler Anforderungen helfen können, wie beispielsweise Brainstorming, Interviews, Umfragen und prototypische Entwicklungsansätze, wie Agile oder Scrum. Diese Techniken können helfen, Anforderungen zu identifizieren, zu priorisieren und zu verfeinern, um sicherzustellen, dass sie die Bedürfnisse der Benutzer und Stakeholder erfüllen.

## Leitfaden und Frameworks für einen erfolgreichen Workshop

### Vorbereitung:

1. Definiere die Ziele der Workshops. Was möchtest du erreichen?
2. Wähle die Teilnehmer aus. Wer muss anwesend sein, um die Anforderungen zu verstehen und zu validieren?
3. Vereinbare einen Termin und Ort für die Workshops. Stelle sicher, dass alle Teilnehmer verfügbar sind.
4. Wähle ein Workshop-Framework aus.
5. Bereite Materialien wie Stift und Papier und Präsentationsfolien vor.

### Workshop mit Stakeholdern:

1. Begrüße die Teilnehmer und stelle die Ziele des Workshops vor.
2. Führe eine Präsentation oder Demonstration der aktuellen Hotelsoftware durch.
3. Stelle offene Fragen, um ein Verständnis für die Anforderungen der Stakeholder zu bekommen.
4. Erstelle eine Liste von Anforderungen, die während des Workshops identifiziert wurden, und teile diese den Teilnehmern zur Überprüfung und Validierung mit.
5. Diskutiere die Anforderungen mit den Stakeholdern und sammle ihr Feedback.
6. Fasse die wichtigsten Punkte zusammen und vereinbare einen nächsten Termin, falls weitere Diskussionen oder Validierung erforderlich sind.

### Abschluss:

1. Überprüfe, ob alle Anforderungen identifiziert und validiert wurden.
2. Erstelle eine Zusammenfassung der Ergebnisse und teile sie allen Teilnehmern mit.
3. Überprüfe die nächsten Schritte und vereinbare einen Termin für die nächste Phase des Projekts.

Es gibt verschiedene Frameworks und Methoden, die bei der Durchführung von Workshops zur Analyse von Anforderungen hilfreich sein können. Hier sind zwei Beispiele:

- [Design Thinking](#): Dies ist ein innovativer, kreativer und nutzerzentrierter Ansatz zur Lösung von Problemen und zur Entwicklung neuer Ideen. Es ist ein Prozess, der sich auf das Verständnis der Bedürfnisse und Herausforderungen von Benutzern konzentriert und darauf abzielt, menschenzentrierte Lösungen zu schaffen. Design Thinking besteht aus verschiedenen Phasen, die sich auf das Verständnis der Benutzerbedürfnisse, die Ideenfindung, Prototyping und Testen der Lösungen konzentrieren. Dabei wird iterativ vorgegangen und die Lösungen werden kontinuierlich verbessert, basierend auf dem Feedback der Benutzer.
- [EventStorming](#): Dies ist eine kollaborative Methode, die im Rahmen des Domain-driven Designs verwendet wird, um komplexe Geschäftsprozesse oder Softwareanforderungen zu verstehen und zu modellieren. Die Methode wurde von Alberto Brandolini entwickelt und basiert auf dem Prinzip der ereignisgetriebenen Architektur. Im EventStorming-Workshop arbeiten verschiedene Stakeholder wie Entwickler, Business-Experten, Designer oder Projektmanager gemeinsam an der Modellierung eines bestimmten Prozesses oder Systems. Die Methode zielt darauf ab, ein gemeinsames Verständnis des Systems zu schaffen, indem es auf einer großen Tafel oder einem Whiteboard visualisiert wird.

## Zusammenfassung

Funktionale Anforderungen beschreiben die funktionalen Eigenschaften eines Produkts oder einer Dienstleistung. Sie beschreiben, welche Funktionen und Prozesse ein System haben soll und wie es arbeiten soll, um die Anforderungen der Benutzer zu erfüllen. Funktionalen Anforderungen sind messbar und präzise und dienen als Basis für die Entwicklung von Software und Produkten.

User Stories und Use Cases sind Techniken, um Anforderungen aus Sicht des Benutzers zu beschreiben. User Stories sind kurze Beschreibungen von Benutzeraktivitäten, die das Verhalten eines Systems beschreiben. Use Cases hingegen sind detailliertere Beschreibungen von Benutzerinteraktionen mit einem System, die den Ablauf eines bestimmten Prozesses beschreiben.

Workshops sind eine wichtige Methode, um Anforderungen zu analysieren und zu sammeln. Workshops können mit verschiedenen Gruppen durchgeführt werden, wie zum Beispiel mit Stakeholdern oder dem Hotelpersonal. In Workshops können verschiedene Methoden und Frameworks wie Design Thinking oder EventStorming verwendet werden, um die Anforderungen und Bedürfnisse der Benutzer zu verstehen und zu definieren.

Insgesamt sind funktionale Anforderungen, User Stories, Use Cases und Workshops wichtige Instrumente, um die Bedürfnisse der Benutzer zu verstehen und in konkrete Anforderungen für Produkte oder Dienstleistungen umzusetzen. Der Einsatz von Frameworks und Methoden wie Design Thinking oder EventStorming kann helfen, die Effektivität und Effizienz der Workshops zur Anforderungsanalyse zu verbessern.

## Risikomanagement

Risikomanagement ist ein unverzichtbarer Bestandteil jeder Projektplanung. Es ist wichtig, mögliche Risiken frühzeitig zu identifizieren, zu bewerten und entsprechende Maßnahmen zu ergreifen, um das Projektziel zu erreichen und unvorhergesehene Kosten oder Verzögerungen zu vermeiden. Im Kontext der Einführung einer Hotelsoftware können verschiedene Risiken auftreten, die es zu berücksichtigen gilt. Ein Beispiel wäre die Gefahr von Systemausfällen oder Datenverlusten, die zu Betriebsunterbrechungen und erheblichen finanziellen Verlusten führen können. Ein weiteres Beispiel wäre die Unzufriedenheit der Mitarbeiter und Kunden aufgrund unzureichender Schulung oder mangelnder Anpassungsfähigkeit der neuen Software. Es ist daher von großer Bedeutung, potenzielle Risiken zu identifizieren und Maßnahmen zur Risikominimierung zu ergreifen, um das Projekt erfolgreich abzuschließen.

Das Risikomanagement spielt bei der Einführung einer Software eine entscheidende Rolle, um potenzielle Risiken zu identifizieren und zu minimieren. Eine Möglichkeit, das Risiko von Unzufriedenheit der Mitarbeiter in Bezug auf mangelnde Anpassungsfähigkeit der neuen Software zu minimieren, besteht darin, eine Schulung und Unterstützung für die Mitarbeiter bereitzustellen. Der Hotelsoftware-Anbieter kann hierbei hilfreich sein, indem er Schulungen, Software-Präsentationen und Demos zur Verfügung stellt, um den Mitarbeitern die Möglichkeit zu geben, die Software besser kennenzulernen und zu verstehen. Durch eine umfassende Schulung können die Mitarbeiter befähigt werden, die neue Software effektiv zu nutzen und sich an Veränderungen anzupassen, was letztendlich zu einem reibungslosen Übergang führt.

## Wie wird Risikomanagement betrieben?

Risikomanagement kann auf unterschiedliche Weise betrieben werden, abhängig von der Art und Größe des Projekts sowie den Risiken, die berücksichtigt werden müssen. Einige gängige Methoden sind die Identifikation von Risiken, die Bewertung von Risiken und die Entwicklung von Strategien zur Risikominimierung.

Die Formulierung von Risiken erfolgt in der Regel in Form von Risikoanalysen und Risikobewertungen. Hierbei werden potenzielle Risiken identifiziert, ihre Wahrscheinlichkeit und Auswirkungen auf das Projekt bewertet und Prioritäten festgelegt.

Um Risiken festzuhalten, können verschiedene Tools und Techniken verwendet werden, wie z.B. Risikoregister oder Risikomatrixen. Diese dokumentieren die Risiken, ihre Bewertungen und die Strategien zur Risikominimierung. Ein Risikoregister kann z.B. Informationen zu jedem Risiko enthalten, wie z.B. die Beschreibung des Risikos, seine Auswirkungen, seine Wahrscheinlichkeit, wer für die Überwachung des Risikos verantwortlich ist und welche Maßnahmen zur Minimierung des Risikos ergriffen werden sollen.

Ein Beispiel für ein Risiko bezüglich unzufriedener Mitarbeiter im Kontext der Einführung einer Hotelsoftware könnte lauten: "Mangelnde Schulung und Unterstützung der Mitarbeiter bei der Nutzung der neuen Software kann zu Unzufriedenheit und geringer Produktivität führen."

### Beispiel Risiko in einem Risikoregister

Risiko: Unzufriedenheit der Mitarbeiter bezüglich mangelnder Anpassungsfähigkeit der neuen Software

Beschreibung: Es besteht das Risiko, dass Mitarbeiter aufgrund mangelnder Schulung und Vorbereitung auf die neue Software unzufrieden werden und sich gegen die Einführung wehren.

Eintrittswahrscheinlichkeit: Mittel

Auswirkungen: Hoch

Risikobewertung: Kritisch

Risikobewältigung: Minimierung durch Demos und Schulungen: Der Hotelsoftware-Anbieter kann Schulungen anbieten, um den Mitarbeitern die Anpassung an die neue Software zu erleichtern und deren Zufriedenheit zu erhöhen. Durch Demos können die Mitarbeiter die Vorteile der neuen Software erkennen und sich auf die Umstellung vorbereiten.

## Zusammenfassung

Ein Risikomanagement-Prozess kann dazu beitragen, potenzielle Risiken zu identifizieren und deren Auswirkungen auf das Projekt zu minimieren. Risiken können in einem Risikoregister dokumentiert werden, welches eine Liste aller identifizierten Risiken und deren Bewertung beinhaltet. Eine Möglichkeit, das Risiko der Unzufriedenheit der Mitarbeiter aufgrund mangelnder Anpassungsfähigkeit der neuen Software zu minimieren, ist durch Schulungen und Trainings. Ein Schulungsprogramm kann dazu beitragen, dass Mitarbeiter die neuen Prozesse und Funktionen besser verstehen und somit produktiver und zufriedener mit der neuen Software arbeiten können. Insgesamt hilft ein effektives Risikomanagement dabei, die Wahrscheinlichkeit und Auswirkungen von Risiken im Projekt zu reduzieren und somit den Erfolg der Einführung einer Hotelsoftware zu sichern.

# Lastenheft für die Einführung einer Hotelsoftware

## Zielsetzung

Das Ziel dieses Projekts ist es, eine neue Hotelsoftware erfolgreich in unserem Unternehmen zu implementieren. Die neue Software soll dabei helfen, unsere Prozesse zu optimieren und unsere Effizienz zu steigern. Die Einführung soll innerhalb von 6 Monaten abgeschlossen sein.

## Stakeholder

Die folgenden Stakeholder sind an diesem Projekt beteiligt:

Hotelmanager: Verantwortlich für die Überwachung des Projekts und die Genehmigung von Änderungen

IT-Abteilung: Verantwortlich für die technische Umsetzung der Einführung

Mitarbeiter: Werden mit der neuen Software arbeiten und müssen entsprechend geschult werden

Gäste: Werden von der Einführung der neuen Software betroffen sein und sollten daher einen reibungslosen Übergang erfahren

## Funktionale Anforderungen

### User-Story (Option-1)

Als Mitarbeiter im Hotel benötige ich eine benutzerfreundliche Software, um schnell und einfach Reservierungen durchzuführen.

### Use-Case (Option-2)

Use-Case-Name: Reservierung durchführen

Beschreibung: Ein Mitarbeiter führt eine Reservierung über die neue Hotelsoftware durch.

Akteure: Mitarbeiter

Vorbedingung: Mitarbeiter hat sich in der Software angemeldet

Nachbedingung: Reservierung wurde erfolgreich durchgeführt

Prozessschritte:

1. Mitarbeiter öffnet die Software und wählt die Option "Reservierung durchführen"
2. Mitarbeiter gibt die notwendigen Informationen ein, wie z.B. Name des Gastes, Check-in-Datum, Check-out-Datum, Zimmerkategorie
3. Mitarbeiter bestätigt die Reservierung und speichert die Daten

## Nicht-funktionale Anforderungen

- Die neue Software muss intuitiv und benutzerfreundlich sein, um die Einarbeitungszeit zu minimieren.
- Die Software muss sicher und zuverlässig sein, um die Daten unserer Gäste zu schützen.
- Die Software muss skalierbar sein, um zukünftiges Wachstum zu unterstützen.

## Risikomanagement

Risiko: Unzufriedenheit der Mitarbeiter aufgrund mangelnder Anpassungsfähigkeit der neuen Software

Auswirkungen: Verzögerungen bei der Einführung, geringere Effizienz bei der Nutzung der neuen Software

Wahrscheinlichkeit: Hoch

Maßnahmen: Der Hotelsoftware-Anbieter wird Schulungen und Demos zur Verfügung stellen, um die Mitarbeiter auf die neue Software vorzubereiten. Die Schulungen werden vor dem Einsatz der Software durchgeführt, um sicherzustellen, dass die Mitarbeiter über ausreichende Kenntnisse verfügen.