Pflichtenheft

<version>

<projekt>

Verteiler:

**Name (alphab.) Abteilung Ort**

<name> <abteilung> <ort>

<name> <abteilung> <ort>

Dokumentenverwaltung

**Dokument-Historie**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Status** | **Datum** | **Verantwortlicher** | **Änderungsgrund** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Änderungsberechtigte**

<name> <abteilung> <ort>

<name> <abteilung> <ort>

**Dokument wurde mit folgenden Tools erstellt:**

WINWORD 6.0

<Grafiktool>

Inhaltsverzeichnis

1 Einleitung

1.1 Zweck des Dokuments

1.2 Gültigkeit des Dokuments

1.3 Begriffsbestimmungen und Abkürzungen

1.4 Zusammenhang mit anderen Dokumenten

1.5 Überblick über das Dokument

2 Allgemeine Beschreibung des Produkts

2.1 Zusammenhang mit bereits laufenden Projekten

2.2 Zusammenhang mit Vorgänger- und Nachfolgeprojekten

2.3 Zweck des Produkts

2.4 Abgrenzung und Einbettung des Produkts

2.5 Überblick über die geforderte Funktionalität

2.6 Allgemeine Einschränkungen

2.7 Vorgaben zu Hardware und Software

2.8 Benutzer des Produkts

3 Detaillierte Beschreibung der geforderten Produktmerkmale

3.1 Lieferumfang

3.2 Abläufe (Szenarien) von Interaktionen mit der Umgebung

3.3 Ziele des Benutzers

3.4 Geforderte Funktionen des Produkts

3.4.1 <Funktion Bezeichnung a>

3.4.1.1 Wirkungsweise von <Funktion Bezeichnung a>

3.4.1.2 Abhängigkeiten / Randbedingungen

3.4.2 <Funktion Bezeichnung b>

3.5 Externe Schnittstellen des Produkts

3.5.1 Benutzerschnittstellen (User Interfaces)

3.5.2 Systemschnittstellen

3.5.2.1 <Schnittstelle a>

3.5.2.2 <Schnittstelle b>

3.6 Sonstige geforderte Produktmerkmale

3.6.1 Geschwindigkeitsmerkmale (performance)

3.6.2 Ressourcenmerkmale (resource)

3.6.3 Schutzmerkmale (security)

3.6.4 Sicherheitsmerkmale (safety)

3.6.5 Portabilitätsmerkmale (portability)

3.6.6 Stabilitätsmerkmale (reliability)

3.6.7 Wartungsmerkmale (maintenance)

3.6.8 Wiederverwendbarkeitsmerkmale (reuse)

3.6.9 Benutzbarkeitsmerkmale (usability)

4 Vorgaben an die Projektabwicklung

4.1 Anforderungen an die Realisierung

4.2 Fertige und zugekaufte Komponenten

4.3 Unterauftragnehmer

4.4 Abnahmebedingungen

4.5 Lieferbedingungen

4.6 Anforderungen an den Einsatz

4.7 Gewährleistung

5 Verpflichtungen des Auftraggebers

6 Literaturverweise

7 Anhang

Allgemeine Vorbemerkung: Dieses kommentierte Inhaltsverzeichnis eines Pflichtenhefts bezieht sich auf Projekte, die sich schwerpunktmäßig mit SW-Entwicklung beschäftigen.

# Einleitung

## Zweck des Dokuments

Der Zweck des vorliegenden Pflichtenhefts ist eine für die Entwicklung verbindliche und möglichst eindeutige Spezifikation von <xyz>. In diesem Sinn enthält es die Summe aller aus Projektsicht erforderlichen und akzeptierten Anforderungen an dieses Produkt und die Projektabwicklung.

## Gültigkeit des Dokuments

Für welchen Bereich ist das Pflichtenheft gültig?

* Gilt das Pflichtenheft für ein konkretes gesamtes Projekt?
* Gilt das Pflichtenheft lediglich für ein Teilprojekt?
* Setzt das Pflichtenheft bereits auf ein bestehendes Pflichtenheft auf?
* Setzt das Pflichtenheft bereits auf ein bestehendes Produkt auf (neue Version, Deltapflichtenheft etc.)?

Hier kann sinnvollerweise auch etwas über die Pflege des Dokuments ausgesagt werden (wer ist für Änderungen zuständig, z.B. bei Versionsentwicklung oder Wartungsprojekt?).

## Begriffsbestimmungen und Abkürzungen

Auflistung von Begriffsdefinitionen und Abkürzungen, die für das Verständnis des Pflichtenhefts wichtig sind (EDV-Fachausdrücke und Begriffe aus der Anwendungsdomäne).

Diese Auflistung wird normalerweise alphabetisch geordnet sein, sie kann aber ggf. auch hierarchisch organisiert werden, sodaß speziellere Begriffe (wie z.B. Transaktion mit einem Bankomaten) bei den entsprechend allgemeineren Begriffen (wie z.B. Transaktion) zu finden sind.

Die Begriffsdefinitionen sind deshalb so besonders wichtig, weil das Pflichtenheft die gemeinsame Verständnisbasis von Auftraggeber und Auftragnehmer darstellt.

Wenn diese Begriffe im Fließtext vorkommen, dann sollen sie typographisch hervorgehoben werden, um ihre Verwendung als Fachausdruck zu kennzeichnen.

## Zusammenhang mit anderen Dokumenten

Sofern kein Angebot vorliegt, in dem alle Anforderungen rechtsverbindlich beschrieben sind, stellt das Pflichtenheft die zwischen Auftraggeber und Auftragnehmer abgestimmte verbindliche Grundlage der Entwicklung und die Basis für die Produktabnahme dar. Somit ist es in diesem Fall für Auftraggeber und Auftragnehmer in gleicher Weise relevant.

Wenn allerdings ein rechtsverbindliches Angebot existiert, ist das Pflichtenheft im allgemeinen nur für die Entwicklung intern relevant (sofern nicht vertraglich anders geregelt).

Bei allen Aussagen, welche die Produktmerkmale, die Projektabwicklung und die Verpflichtungen des Auftraggebers betreffen, muß auf Übereinstimmung mit dem Angebot geachtet werden. Keine diesbezüglichen Aussagen im Pflichtenheft dürfen denen im Angebot widersprechen.

Im Fall, daß ein Lastenheft existiert, muß sich das Pflichtenheft weitestgehend auf dieses beziehen, um die Verfolgbarkeit der dort aus Auftraggeber- bzw. Benutzersicht beschriebenen Anforderungen zu den Anforderungen an das zu erstellende Produkt und die Projektabwicklung zu gestatten. Im Gegensatz zum Lastenheft muß das Pflichtenheft genau spezifizieren, was ein Produkt leisten können muß. Durch diese Spezifikation im Pflichtenheft wird festgelegt, auf welche Weise durch das zu erstellende Produkt die Anforderungen aus dem Lastenheft erfüllt werden.

## Überblick über das Dokument

* Welche Inhalte enthält der Rest des Pflichtenhefts?
* Wie ist das Pflichtenheft aufgebaut?

# Allgemeine Beschreibung des Produkts

Diese allgemeine Beschreibung des zu erstellenden Produkts soll ein „Management Summary“ geben, also noch keine spezifischen Produktmerkmale beschreiben.

## Zusammenhang mit bereits laufenden Projekten

Falls es Zusammenhänge mit bereits laufenden Projekten gibt, so sollen sie hier kurz dargestellt werden (z.B. Überprojekte, Schwesterprojekte). Nötige Schnittstellen sollen dann in Kap. 2.5 bzw. bei der ausführlichen Beschreibung der Systemschnittstellen beschrieben werden.

## Zusammenhang mit Vorgänger- und Nachfolgeprojekten

Falls es Zusammenhänge mit Vorgänger- und Nachfolgeprojekten gibt, so sollen sie hier kurz dargestellt werden. Falls das Produkt ein früheres Produkt ersetzen wird, dann sollen in Kap. 2.5. die Gründe für die Neuentwicklung beschrieben werden.

## Zweck des Produkts

Welcher Zweck wird mit diesem Produkt verfolgt, was sind die wichtigen Merkmale und die entscheidenden Vorteile gegenüber der bisherigen Situation?

Falls das Produkt ein früheres Produkt ersetzen wird, dann sollen hier die Gründe für die Neuentwicklung beschrieben werden.

## Abgrenzung und Einbettung des Produkts

Hier soll die Abgrenzung und Einbettung des zu erstellenden Produkts in seine Umgebung beschrieben werden.

Welche wesentlichen Leistungen soll das zu erstellende Produkt umfassen und welche nicht, bzw. was gehört nicht dazu sondern zur Umgebung?

Diese Beschreibung der Systemabgrenzung muß konsistent sein mit evtl. vorhandenen übergeordneten Dokumenten.

Sollten für den Fall, daß ein Lastenheft existiert, diesem gegenüber wesentliche Einschränkungen in der Funktionalität vorgenommen werden, so müssen diese hier beschrieben werden.

Welche Beziehungen hat dieses Produkt zu anderen Systemen in seiner Umgebung (z.B.: Handelt es sich um ein unabhängiges System oder ein Subsystem eines größeren Systems; ersetzt es ein bereits bestehendes System?).

Wenn es sich bei dem zu erstellenden Produkt um ein Subsystem handelt, sollen hier die Grundzüge des übergeordneten Systems sowie die Beziehungen zu den anderen Subsystemen zusammengefaßt werden (z.B. in Form eines Blockdiagramms, das die Komponenten, deren Zusammenspiel und die externen Schnittstellen zeigt).

## Überblick über die geforderte Funktionalität

Hier soll ein Überblick darüber gegeben werden, was das zu erstellende Produkt seiner Umgebung gegenüber (z.B. menschlichen Benutzern oder anderen Systemen) leisten soll.

Diese Beschreibung soll gegenüber der detaillierten Produktbeschreibung in Kap. 3 lediglich einen Überblick vermitteln.

## Allgemeine Einschränkungen

Hier sollen die wesentlichen Vorgaben beschrieben werden, welche die Entwicklung und das Produkt einschränken (z.B. Vorgaben hinsichtlich Schnittstellen, Standards, Methoden).

## Vorgaben zu Hardware und Software

* In welcher HW-Umgebung soll das Zielsystem laufen?
* In welcher SW-Umgebung soll das Zielsystem laufen?
* In welcher HW-Umgebung soll das System entwickelt werden?
* In welcher SW-Umgebung soll das System entwickelt werden (Betriebssysteme, Entwicklungstools, etc.)?

## Benutzer des Produkts

Hier soll charakterisiert werden, welche Benutzer das zu erstellende Produkt einsetzen werden (z.B. Ausbildung und Vorkenntnisse bei der Anwendung von Software und Computern).

Dabei ist ggf. zu beachten, daß es verschiedene Benutzerklassen geben kann (z.B. produktive Nutzer und Systemadministratoren; tägliche und fallweise Benutzer).

# Detaillierte Beschreibung der geforderten Produktmerkmale

In diesem Kapitel des Pflichtenhefts müssen die vom Auftraggeber geforderten (vorgeschriebenen) und von der Entwicklung zugesagten Produkteigenschaften definiert sein. Diese Eigenschaften beinhalten alle Funktionen, Schnittstellen und sonstigen Produktmerkmale. Dabei sollten bei der Beschreibung der Eigenschaften einige Punkte beachtet werden:

Jedes geforderte Merkmal sollte:

* eindeutig bezeichnet sein (Bezeichnung / Identifier)
* bei Bedarf priorisiert werden (Priorität / Priority), etwa in Hinsicht auf Ausbaustufen
* die Herkunft der Anforderung beinhalten (Ursprung / Origin), sofern es dabei Unklarheiten geben kann, wer etwas fordert
* sofern ein Lastenheft vorliegt, den Bezug zu den entsprechenden Anforderungen in diesem (möglichst mit Verweis auf eine eindeutige Bezeichnung) beinhalten (Quelle / Source);  
  wenn nicht zu allen Anforderungen im Lastenheft Bezüge herstellbar sind (im Falle von Einschränkungen der Funktionalität), so sollten auch Verweise auf solche Anforderungen im Lastenheft mit entsprechenden Hinweisen gemacht werden.

Jedes geforderte Merkmal sollte so beschrieben sein, daß:

* möglichst geringer Interpretationsspielraum bleibt (Klarheit / Clarity)
* Überprüfbarkeit gegenüber dem fertiggestellten Produkt gegeben ist (gemäß der Frage: Ist das Produkt richtig gemacht?)

## Lieferumfang

Im Lieferumfang muß eine Liste aller Lieferkomponenten enthalten sein (Produkt und dessen Teilkomponenten, auszuliefernde Projektdokumentation, Benutzerhandbücher, Schulungsunterlagen, Training, Einführungsunterstützung z.B. in Form von „Coaching“, Installationsprozeduren, Migration von Datenbeständen, Benutzer Support in Form von Hotline oder Email, etc. ...).

Jene Komponenten, die nicht das „Produkt im engeren Sinne“ (Programm) darstellen, wie z.B. Benutzerdokumentation, Training usw. sind in diesem Unterkapitel genau zu beschreiben; das „Produkt im engeren Sinne“ wird ausführlich in den folgenden Unterkapiteln beschrieben.

## Abläufe (Szenarien) von Interaktionen mit der Umgebung

Hier sollen typische Abläufe beschrieben werden, wie ein (oder mehrere) Benutzer bzw. ein anderes System in der Umgebung des zu erstellenden Produkts mit diesem interagieren. Solche Abläufe werden oft auch als Szenarien bezeichnet.

Wenn derartige Szenarien in einem ggf. existierenden Lastenheft bereits ausführlich beschrieben sind, können sie von dort übernommen und vervollständigt werden.

Es ist zwar im Normalfall nicht möglich, alle möglichen Szenarien vollständig zu beschreiben. Allerdings muß eine gewisse Vollständigkeit erreicht werden in bezug auf wichtige Zusammenhänge mit den unten zu beschreibenden Funktionen des zu erstellenden Produkts:

1. Alle vom Produkt durchzuführenden Aktionen müssen mittels entsprechender Funktionen realisiert werden können. Auf diese sollte so genau wie möglich und explizit verwiesen werden (z.B. durch eine Angabe der Art „durch Funktion <a> in 3.4.x“). Damit soll erreicht werden, daß möglichst keine Funktionen vergessen werden.
2. Jede der unten zu beschreibenden Funktionen sollte in mindestens einem Ablauf vorkommen. Damit soll erreicht werden, daß möglichst keine Funktionen gefordert werden, von denen man eigentlich gar nicht weiß, wozu man sie benötigt.

Analog dazu wird empfohlen, bei allen von Benutzern oder anderen Systemen in der Umgebung durchzuführenden Aktionen Verweise auf die entsprechenden unten zu beschreibenden Schnittstellen zu verweisen.

Ebenso muß eine gewisse Vollständigkeit erreicht werden in bezug auf wichtige Zusammenhänge mit den unten zu beschreibenden Zielen des Benutzers:

1. *Alle Szenarien sollten ein oder mehrere Ziele verfolgen. Wenn ein solches nicht unmittelbar selbstverständlich ist (etwa bei einfachen und kurzen Szenarien), dann sollte so genau wie möglich und explizit verwiesen werden. Dadurch wird das Verständnis der Szenarien erleichtert.*
2. Jedes der unten zu beschreibenden Ziele des Benutzers sollte durch mindestens einen Ablauf erreicht werden können. Dadurch soll erkennbar sein, daß diese Ziele mit Hilfe des zu erstellenden Produkts auch erfüllbar sind.

## Ziele des Benutzers

*Hier sollen alle Ziele des Benutzers beschrieben werden, welche durch die in den Szenarien oben beschriebenen Interaktionen mit dem Produkt erreicht werden (z.B. wird durch die Interaktionen mit einem Geldausgabeautomaten im Normalfall das Ziel des Benutzers erfüllt, daß er (mehr) Bargeld hat.). Dadurch wird das Verständnis der Szenarien erleichtert.*

*Im allgemeinen ist es möglich, daß zur Erfüllung eines Zieles mehrere Szenarien geeignet sind bzw. durch einen solchen Ablauf mehrere Ziele erfüllt werden. In jedem Fall sollte so genau wie möglich und explizit auf die entsprechenden Szenarien verwiesen werden.*

*In manchen Fällen kann es sinnvoll sein, auch allgemeiner die Ziele des Auftraggebers zu untersuchen, die über diese unmittelbaren Ziele des Benutzers hinausgehen. Da solche Ziele aber nicht so viel mit den Szenarien zu tun haben, sollten sie getrennt in einem eigenen Unterkapitel bzw. einem Anhang beschrieben werden. Wenn es dadurch ggf. zu Problemen mit dem Auftraggeber kommen könnte, sollten solche Ziele gar nicht im Pflichtenheft beschrieben werden.*

## Geforderte Funktionen des Produkts

Hier muß detailliert beschrieben werden, was das zu erstellende Produkt leisten soll (es geht dabei um eine Beschreibung aus Sicht der Reaktionen des Systems, wobei die Anforderungen aus den Abläufen abgebildet werden müssen).

Diese Beschreibung muß mit dem Überblick über die geforderte Funktionalität (in Kap. 2) übereinstimmen.

Für die Beschreibung eignet sich üblicherweise eine hierarchische Gliederung der Funktionen (Funktionsbaum). Für die hierarchische Gliederung bieten sich grundsätzlich zwei Arten an:

1. Wenn auf allen Ebenen konkrete Funktionen beschrieben werden, geht es darum, die Teilfunktionen den übergeordneten zusammengesetzten Funktionen zuzuordnen.  
   (z.B. setzt sich die konkrete Funktion „Identifizierung eines Bankkunden“ (unter anderem) aus den konkreten Funktionen „Aufforderung zum Eingeben eines Codes“ und „Überprüfen eines eingegebenen Codes“ zusammen.)
2. Wenn nur auf unterster Ebene konkrete Funktionen beschrieben werden, dann geht es darum, Klassen solcher Funktionen zu definieren und die konkreten Funktionen den entsprechenden Klassen zuzuordnen.  
   (z.B. enthält die Klasse von „Funktionen zur Identifizierung eines Bankkunden“ durch einen Bankomaten die konkreten Funktionen „Aufforderung zum Eingeben eines Codes“ und „Überprüfen eines eingegebenen Codes“.)

Wenn eine zusammengesetzte Funktion (wie in diesem Beispiel „Identifizierung eines Bankkunden“) bei Ihrer Durchführung Interaktionen mit dem Benutzer erfordert (hier das Eingeben des Codes), dann ist es möglicherweise günstiger, die hierarchische Gliederung gemäß der zweiten Art mittels Klassen von Funktionen durchzuführen. In diesem Fall sollte jede solche Klasse (im Beispiel „Funktionen zur Identifizierung eines Bankkunden“) einen eigenen Unterabschnitt des Dokuments ausmachen, in dem die zugehörigen konkreten Funktionen beschrieben werden. Grundsätzlich ist es auch möglich und oft sinnvoll, zu solchen Klassen von Funktionen speziellere Unterklassen zu definieren, um eine tiefergehende hierarchische Gliederung zu erreichen.

Jede Beschreibung einer konkreten Funktion muß eine eindeutige Bezeichnung, eine Beschreibung (Wirkungsweise nach außen) und, falls vorhanden, Abhängigkeiten und Randbedingungen enthalten.

### <Funktion Bezeichnung a>

Jede Funktion muß eine eindeutige Bezeichnung haben. Diese Bezeichnung sollte im Lebenslauf des Projekts (also Entwurf, Realisierung, usw.) wiedergefunden werden.

#### Wirkungsweise von <Funktion Bezeichnung a>

Hier muß die Wirkungsweise der Funktion beschrieben werden, im Sinne des Effekts, den das zu erstellende Produkt mittels dieser Funktion auf seine Umgebung hat. Dies sollte in jedem Fall so detailliert geschehen, daß nur eine (1) Interpretation der Wirkungsweise der Funktion möglich ist.

#### Abhängigkeiten / Randbedingungen

Alle ggf. existierenden Abhängigkeiten bzw. Zusammenhänge mit anderen Funktionen müssen hier festgehalten werden. Damit soll erreicht werden, daß bei Änderungen einer hier beschriebenen Funktion auch diese abhängigen Funktionen gezielt betrachtet werden können. Darüber hinaus sollen ev. Redundanzen bzw. insbesondere Widersprüche vermieden werden.

Sollte es zu einer hier beschriebenen Funktion wesentliche Randbedingungen geben (wie z.B. bezüglich der Geschwindigkeit des Produkts bei deren Erfüllung, dann müssen diese unten bei den sonstigen Produktmerkmalen beschrieben und von hier aus referenziert werden (z.B. mit einem Verweis auf das entsprechende Geschwindigkeitsmerkmal).

### <Funktion Bezeichnung b>

....

## Externe Schnittstellen des Produkts

Hier müssen alle Benutzerschnittstellen und alle Schnittstellen des zu erstellenden Produkts zu Systemen in seiner Umgebung beschrieben werden.

Das externe Verhalten im Sinne der Interaktionen des zu erstellenden Produkts mit seiner Umgebung sollte bereits oben in Form von Szenarien beschrieben sein. Hier geht es primär darum zu beschreiben, wie die konkreten Schnittstellen aussehen und funktionieren.

### Benutzerschnittstellen (User Interfaces)

Hier muß beschrieben werden, in welcher Form das Produkt mit den Benutzern kommuniziert. Bei Anwendungen mit graphischem User Interface sollte sich die Gliederung an dessen sichtbaren Elementen orientieren.

Für die Beschreibung der Benutzerschnittstellen empfiehlt sich eine Aufteilung in einen allgemeinen Teil mit Hinweisen auf den generellen Aufbau der Benutzerschnittstelle (z.B. „Aufteilung in Menü/Bild/Statuszeile“, „wie sind Help-Seiten aufgebaut?“, „Verwendung von OK/Abbruchbutton“ usw.) und in spezielle Teilbeschreibungen mit den einzelnen Teilen der Benutzerschnittstelle (z.B. Eingabemasken, Menüs, usw.).

Es sollte hier allerdings nicht beschrieben werden, wie die Benutzerschnittstelle intern realisiert wird. Etwaige Vorgaben des Aufraggebers bezüglich zu verwendender Bibliotheken oder „GUI Builder“ sollten unten in Kap. 4 beschrieben werden.

Außerdem können hier Verweise auf Prototypen oder Lösungsstudien stehen (immer häufiger werden Benutzerschnittstellen ja nicht „auf dem Papier“ definiert).

Beispiele dafür, was zu beschreiben ist:

* Eingabedaten
* Dialog, Benutzerführung
* Kommandos, Steuerparameter
* Menüs
* Eingabemasken
* Grafik
* Ausgabedaten
* Protokolle, Listen
* Betriebsmeldungen
* Fehlerreaktionen
* Helptexte
* Bedienelemente (Anordnung, Funktion)
* Anzeigen und Signalisierung (Anordnung, Funktion)

### Systemschnittstellen

Hier müssen alle Schnittstellen zu Software- und Hardwaresystemen beschrieben werden, mit denen das zu erstellenden Produkt kommunizieren wird. Wesentlich dabei ist, daß jede solche Systemschnittstelle neben einer entsprechenden Bezeichnung auch eine Beschreibung der Kommunikationsart, des Datenformats und anderer Details enthält.

Ein grober Überblick sollte bereits in Kap. 2 dargestellt sein. Falls dieser aufgrund der Komplexität der Systemschnittstellen für deren genaues Verständnis nicht ausreichen, muß hier noch eine detailliertere Übersicht gegeben werden. In jedem Fall muß gewährleistet sein, daß die Zusammenhänge der im folgenden beschriebenen einzelnen Schnittstellen klar werden.

#### <Schnittstelle a>

* Syntax/Semantik der Daten der Schnittstelle
* Datenformat
* Programmschnittstelle
* Übertragungsprotokolle
* Datenraten

#### <Schnittstelle b>

....

## Sonstige geforderte Produktmerkmale

Es geht hier um weitere Produktmerkmale (oft auch als Qualitätsmerkmale bezeichnet), die das Produkt genauer kennzeichnen und über Funktionen und Schnittstellen hinausgehen (alle zusätzlich zu erfüllenden Anforderungen). Um zu entscheiden, was hier wirklich niedergeschrieben werden muß, sollte man die unterschiedlichen Gründe für eine Beschreibung bedenken:

* Explizite Anforderungen des Auftraggebers, deren Erfüllung hier versprochen wird.
* Weitere wichtige Produktmerkmale, auch wenn sie nicht explizit gefordert wurden (dies dient ganz wesentlich zur Absicherung beider Seiten, um Fehlinterpretationen und falsche Hoffnungen zu vermeiden): Die wenigsten Dinge sind wirklich so „selbstverständlich“, wie man es sich in der eigenen Sichtweise vorstellt.

Falls zu einem Teilkapitel keine Anforderungen vorliegen, so soll dies ebenfalls dokumentiert werden (z.B. „Keine besonderen Sicherheitsmerkmale gefordert“).

### Geschwindigkeitsmerkmale (performance)

Hier werden jene Eigenschaften beschrieben, welche die Geschwindigkeit des Produkts definieren (geforderte Werte). Dabei sind Standardformulierungen wie „rasche Antwortzeiten werden gefordert“ zu vermeiden. Vielmehr sind exakte Werte einzusetzen.

Einige Beispiele für Geschwindigkeitsmerkmale sind:

* RealTime, Time-Sharing, Batch
* Reaktionszeiten/Antwortzeiten
* Anlaufzeiten
* Durchsatzrate
* Belegungsdauer

### Ressourcenmerkmale (resource)

Hier werden Merkmale beschrieben, die angeben, mit welchem Bedarf an Ressourcen das zu erstellende Produkt auskommen muß.

Dabei sind unexakte Formulierungen (siehe oben) zu vermeiden (Interpretationsspielraum!)

Einige Beispiele für Ressourcenmerkmale sind:

* Datenmengen
* CPU-Bedarf
* CPU-Auslastung
* Hauptspeicher
* Peripheriespeicher
* Peripheriegeräte
* Ausgabemenge (z.B. Papier)
* Benötigtes Bedienpersonal (läuft das System unbedient oder muß jemand eventuelle Meldungen beantworten?)

### Schutzmerkmale (security)

Hier müssen jene Merkmale beschrieben werden, die dem Schutz des Produkts gegen Eingriffe von außen dienen (gegen unberechtigten Zugriff, Integritätsverletzungen durch gleichzeitigen Dateizugriff, Einschränkungen der Lesezugriffe, Virenschutz usw.).

### Sicherheitsmerkmale (safety)

Hier müssen alle Eigenschaften festgehalten werden, die der Sicherheit dienen. Damit sind jene Merkmale gemeint, welche die Schadensmöglichkeit nach einem Softwarefehler oder Systemausfall begrenzen. Dazu dienen z.B. alle Formen der Datensicherung oder Fehleraufsetzverhalten. Die Überlegungen zu Sicherheitsmerkmalen können hilfreich dabei sein, kritische Softwareteile zu identifizieren.

### Portabilitätsmerkmale (portability)

Hier müssen ggf. alle Merkmale stehen, die eine Adaptierung oder Portierung des Produkts unterstützen, z.B. GUI-Tools, ANSI-C, oder ähnliches. Außerdem sollte - falls gefordert - auch auf alle Punkte, die bei einer Portierung oder Adaptierung von Interesse sind, explizit hingewiesen werden.

### Stabilitätsmerkmale (reliability)

Hier müssen ggf. Eigenschaften definiert werden, die Aussagen über die Zuverlässigkeit und Verfügbarkeit des Produkts ermöglichen. Dies können z.B. sein: Ausfallzeiten in Minuten/Jahr, mittlere Zeit zwischen dem Auftreten von Softwarefehlern (MTBF = MeanTime Between Failure).

### Wartungsmerkmale (maintenance)

Hier müssen ggf. Angaben zur Wartungsfreundlichkeit des Produkts stehen. Neben der Strukturierung von Komponenten, dem Einsatz von CASE-Tools, Design-Tools oder Designmethoden können dies auch Angaben zur mittleren Zeit für eine Softwarekorrektur (MTTR = MeanTime To Repair) sein.

### Wiederverwendbarkeitsmerkmale (reuse)

Hier sind Anforderungen an das Produkt bzw. an Produktteile gemeint, die eine spätere Wiederverwendung ermöglichen (z.B. verstärkte Standardisierung, Modularisierung, Dokumentation etc.). Die projektinterne WV hingegen, wie z.B. Standards (Vorlagen, Strukturen, etc.) und Methoden usw. ist im WV-Plan zu beschreiben, wenn sie nicht ausdrücklich vom Auftraggeber verlangt wird.

### Benutzbarkeitsmerkmale (usability)

*Hier müssen ggf. vom Kunden geforderte oder implizit aufgrund der Benutzerklassen (siehe Kap. 2) zu fordernde Merkmale zur Benutzbarkeit angegeben werden. Dies können z.B. Angaben darüber sein, wieviele „Usability Probleme“ bei einem „Usability Test“ mit tatsächlichen Benutzern im Rahmen der Abnahmetests maximal auftreten dürfen. (Dabei ist auch anzugeben, daß diese tatsächlichen Benutzer etwa aus einer Benutzerklasse mit geringer Erfahrung mit der Benutzung von graphischen Benutzerschnittstellen stammen.)*

# Vorgaben an die Projektabwicklung

Im diesem Kapitel müssen alle Vorgaben an die Abwicklung des Projekts beschrieben werden, die für den Auftraggeber relevant sind. Alle weiteren Vorgaben und Bedingungen müssen im Projektplan, CM-Plan, QS-Plan oder WV-Plan stehen.

Bei allen Aussagen muß auf Übereinstimmung mit einem allfälligen Angebot geachtet werden, d.h. Aussagen im vorliegenden Pflichtenheft dürfen keinesfalls solchen im Angebot widersprechen.

## Anforderungen an die Realisierung

Hier müssen alle Voraussetzungen und Bedingungen für die Realisierung des Produkts im Rahmen der Projektabwicklung aufgeführt werden. Dazu zählen HW, SW, Tools, Methoden usw.

Nachfolgend einige Beispiele mit Erläuterungen:

* Hardware
* Entwicklungsrechner
* Betriebsmittel für die Entwicklung (Drucker, Meßgeräte, Netzwerk, ...)
* Testanlagen
* Software
* Betriebssysteme für die Entwicklungsrechner
* Standardsoftware für die Entwicklung
* Sprachen, Shells, Compiler
* Klassenbibliotheken
* SW-Tools
* Sonstiges
* Entwicklungsmethode
* Entwurfsmethoden
* Beibringen von QS-Nachweisen
* Vertraulichkeitsgrad für Entwicklungstätigkeiten und Dokumente

## Fertige und zugekaufte Komponenten

Hier müssen fertige und zugekaufte Komponenten beschrieben werden, die für das Produkt bzw. die Projektabwicklung notwendig sind.

* Standardsoftware
* Wiederverwendete eigene Software oder Software des Auftraggebers (siehe auch Kap. 5)
* Betriebssysteme
* Fremdentwicklung

## Unterauftragnehmer

Abhängig vom vertraglichen Rahmen müssen Unterauftragnehmer gegenüber dem Auftraggeber offengelegt werden. Alle Unterauftragnehmer und ihre Beiträge zum Produkt bzw. Projekt müssen dokumentiert werden. Eventuell können auch wichtige Liefertermine im Pflichtenheft stehen.

## Abnahmebedingungen

Hier müssen alle Bedingungen aufgezählt werden, die für die Abnahme relevant sind.

Nachfolgend einige Beispiele:

* Rahmenbedingungen
* Wogegen wird abgenommen? (im allgemeinen gegen das Pflichtenheft)
* Wie wird abgenommen? (z.B. im Zuge eines gemeinsamen Abnahmetests beim Kunden, oder SW wird ausgeliefert und durch Kundeneinsatz mit Fehlermeldungen und Fristsetzung abgenommen)
* Wer stellt die Testdaten bereit? Wann sind diese Testdaten beizustellen?
* Wo wird abgenommen? (z.B. beim Kunden, auf Kundenanlage, auf Entwicklungsumgebung)
* Wer unterzeichnet das Abnahmeprotokoll / den Abnahmebericht? (Kunde, Entwicklung)
* Abnahmekriterien
* Festlegung des Abnahmetests  
  Wann ist die Abnahme erfolgreich verlaufen? (z.B. Restfehlerquote)
* Werden die sonstigen Merkmale aus dem Pflichtenheft eingehalten?
* Abnahmeunterlagen (z.B. Testprotokolle)
* Sind zusätzliche Unterlagen für die Abnahme notwendig? (z.B. Nachweis von Reviews mit Reviewprotokollen oder Tests mit Testprotokollen)
* Gutachten, Sicherheitsnachweise
* Sind Gutachten oder Sicherheitsnachweise beizubringen?
* Erfüllung von Vorschriften und Normen
* Hält das Produkt vorgeschriebene Normen und Vorschriften ein?

## Lieferbedingungen

Hier müssen alle Lieferbedingungen stehen. Die Lieferbedingungen sind mit dem Auftraggeber zu vereinbaren. Eine genaue Überprüfung der Lieferbedingungen kann auch sehr hilfreich für das Kapitel „“ (Kap. 5) sein. Nachfolgend sind einige Beispiele für Lieferbedingungen angeführt.

Im wesentlichen sind folgende Fragen zu beantworten:

* Wann werden die einzelnen Lieferkomponenten geliefert? (z.B. Lieferplan mit Terminen. Die Abdeckung aller im Lieferumfang beschriebenen Lieferkomponenten ist zu überprüfen).
* Wie werden die einzelnen Lieferkomponenten geliefert (Lieferform)? (z.B. Elektronisch, auf Papier, auf Band, auf Diskette, in Source-Form oder nur ablauffähiger Code, usw.)
* Was sind die Voraussetzungen für eine Lieferung? (z.B. Dokumente, die bei Lieferung beizubringen sind; dies können Zertifizierung, Inbetriebnahme­bewilligungen, Patente/Lizenzen, QS-Nachweise, usw.).

## Anforderungen an den Einsatz

*Hier müssen alle Anforderungen dokumentiert sein, die während des Einsatzes erfüllt werden müssen. Dies können insbesondere solche Anforderungen sein, die das Vorgehen in dieser Phase betreffen.*

*Beispiele können etwa spezielle Anforderungen an einen Probebetrieb oder die Bereitstellung von Personal während der Inbetriebnahme sein.*

## Gewährleistung

Im Unterkapitel Gewährleistung müssen alle Vereinbarungen dokumentiert sein, die für die Gewährleistung des Produkts relevant sind. Dies sind z.B. Gewährleistungsdauer, Umfang der Gewährleistung, Fehlermeldungs­verfahren usw.

# Verpflichtungen des Auftraggebers

Hier muß alles vollständig aufgezählt werden, wozu der Auftraggeber im Rahmen des Projekts verpflichtet ist.

In Kap. 4 können ggf. bereits implizit solche Verpflichtungen erwähnt sein (z.B. im Zusammenhang mit der für die Projektabwicklung vom Auftraggeber vorgeschriebenen Hardware).

Hier müssen aber trotzdem sämtliche Auftraggeberverpflichtungen kompakt auffindbar sein (z.B., daß die vorgeschriebene Hardware vereinbarungsgemäß vom Auftraggeber bereitgestellt werden muß).

Bei allen Aussagen muß auf Übereinstimmung mit einem allfälligen Angebot geachtet werden, d.h. Aussagen im vorliegenden Pflichtenheft dürfen keinesfalls solchen im Angebot widersprechen.

Bei vielen Punkten ist es unbedingt erforderlich, hier auch Termine anzugeben (da sonst z.B. Terminverzug durch fehlende Beistellungen seitens des Auftraggebers droht).

Nachfolgend einige Beispiele

* Zur Verfügung stellen von Hardware (z.B. Entwicklungsrechner)
* Zur Verfügung stellen von Software (z.B. Tools, Lizenzen, Standard-SW, SW des Auftraggebers)
* Zur Verfügung stellen von Dokumentation (z.B. von bestehenden Systemen, unterzeichnetes Lastenheft, SW-Dokumentation)
* Verpflichtungen in Zusammenhang mit Schulung (z.B. Schulung von Entwicklern durch Auftraggeber; Auftraggeberpersonal, das zu schulen ist)
* Zuständigkeiten und Ansprechpartner beim Auftraggeber (Verantwortliche, Auskunftspersonen und Projektleiter beim Auftraggeber; evtl. beigestellte Entwickler des Auftraggebers, Tester, Projektassistenz, Pilotanwender).
* Voraussetzungen und Organisation von Tests (z.B. Testdaten, Testumgebung zur Verfügung stellen, ...)
* Infrastruktur (z.B. zur Verfügung stellen von Räumen, Rechenzentrum, Leitungen, Betriebszeiten, ...)
* Installationsvoraussetzungen (z.B. vorinstallierte HW, Zugang zu Einrichtungen des Kunden, Verfügbarkeit von fachkundigem Personal mit Arbeitsplan)
* Reaktionszeiten des Auftraggebers auf Anfragen
* Vorgehen, falls Änderungen von Anforderungen gewünscht sind
* Validierung von Oberflächen und Programmteilen während der Entwicklung (bei Prototyping)
* ...

# Literaturverweise

Hier müssen alle im Pflichtenheft zitierten Dokumente angeführt werden.

# Anhang

Falls erforderlich, können hier Materialien eingefügt werden, die nicht in die hier definierte Gesamtstruktur des Pflichtenhefts passen und dennoch zum Pflichtenheft gehören können.

Beispiele dafür sind Modelle. Diese können sowohl textuell als auch in mehr oder weniger formalen Notationen dargestellt sein (etwa in Z, VDM, SDL, Entity-Relationship-Diagrammen und/oder endlichen Automaten). Solche Modelle können „bloß“ dazu dienen, die Anwendungsdomäne und deren Beziehungen zum zu erstellenden Produkt besser zu verstehen. Es ist aber prinzipiell auch möglich und in manchen Fällen im Vergleich zum Aufwand nützlich, mit Modellen die Anforderungen selbst darzustellen (im Sinne: das zu erstellende Produkt soll so sein wie das Modell).