

Pflichtenheft / Lastenheft

Das **Pflichtenheft** (PH) wird in der Definitionsphase vom Entwickler erstellt. Es dient der Definition der Anforderungen an das zu erstellende Produkt. Es hat in der Regel Vertragscharakter. Es ist in einer für den Kunden verständlichen Sprache abgefaßt. Um Streitfälle zu vermeiden, sollten die Leistungen des künftigen Produkts klar abgegrenzt und unmißverständlich beschrieben werden.

Historisch enthielt das PH die komplette Anforderungsdefinition, neuerdings wird viel davon in ein OOA-Modell und einen Oberflächenprototyp ausgelagert.

Das **Lastenheft** ist eine Vorstufe des Pflichtenhefts, die vom Kunden, soweit er dazu fähig ist, erstellt wird und enthält die Wünsche des Kunden an das Produkt.

Mögliches Gliederungsschema eines Pflichtenheftes

- 1 Zielbestimmung
 - 1.1 Muss-Kriterien
 - 1.2 Soll-Kriterien
 - 1.3 Kann-Kriterien
 - 1.4 Abgrenzungskriterien
- 2 Einsatz
 - 2.1 Anwendungsbereiche
 - 2.2 Zielgruppen
 - 2.3 Betriebsbedingungen
- 3 Umgebung
 - 3.1 Orgware
 - 3.2 Hardware
 - 3.3 Software
 - 3.4 Produktschnittstellen
- 4 Funktionalität (Anwendungsfälle)
- 5 Mengengerüst (Balzert: Daten)
- 6 Leistungen
- 7 Benutzungsoberfläche
 - 7.1 Benutzermodell
 - 7.2 Kommunikationsstrategie
 - 7.3 Kommunikationsaufbau
 - 7.4 Benutzerdokumentation
- 8 Qualitätsziele
9. Entwicklungs-Konfiguration (Software, Hardware, Orgware)
10. Einführungsstrategie
11. Anhänge
 - 11.1 Abkürzungen
 - 11.2 Glossar
 - 11.3 Literaturusw.

Die **Zielbestimmung** sollte einen einführenden, zusammenfassenden Charakter haben, und die Vorteile des geplanten Produkts deutlich machen.

Die **Muss-Kriterien** beschreiben, was das Produkt mindestens leisten muss, um einen Nutzen zu erbringen; die **Soll-** und die **Kann-Kriterien** beschreiben weitere Ausbaustufen; die **Abgrenzungskriterien** beschreiben, was durch das Produkt **nicht** geleistet werden soll.

Aus den Angaben zum **Einsatz** lassen sich Anforderungen zur Benutzungsoberfläche und zur Qualität ableiten. Für die **Zielgruppen** müssen die notwendigen Vorkenntnisse angegeben werden, unter **Betriebsbedingungen** z.B. Büro/Fabrik, Betriebszeit, (un)beobachtet.

Die **Umgebung** gibt an, welche Kombination von Hardware, Software und Orgware für die Einsatzfähigkeit des Produktes vorgesehen ist. Immer mit Richtungsangabe: ab/bis/genau. **Bsp.:**

Software: JRE ab 1.5, Oracle ab 9, Windows genau XP SP 2

Hardware: Intel-Server mit Ethernet-Anschluß genau 100Mb, Festplatte ab 50 GB, RAM ab 1 GB

Orgware: bestimmte Formulare und Fonts.

Die **Produktschnittstellen** legen (evtl. als Anhang) mit anderen Softwaresystemen auszutauschende Daten und deren Übergabeprotokolle fest.

Im Hauptteil **Funktionalität** beschreibt man die wesentliche Leistung des Produkts auf oberster Abstraktionsebene. Sammlung typischer Arbeitsabläufe bis zum Ergebnis für den Bediener. Kann die Anwendungsfalldiagramme und die Anwendungsfallspezifikationen enthalten oder auf sie verweisen.

Im **Mengengerüst** (Balzert: Daten) werden **statische** (z.B. bis 10.000 Leser im Bestand) und **dynamische** (z.B. bis 1.000 Ausleihen/Tag) mengenmäßige Zusagen gegeben. Unter **Leistungen** werden zeitliche Zusagen gegeben (z.B. bis 2 Sekunden für eine Ausleihe)

Bei der Beschreibung der **Benutzungsoberfläche** können auch andere Software-Produkte als „Benutzer“ aufgefaßt werden.

Das **Benutzermodell** gibt an, welche Benutzergruppen (Anfänger, Experten, Gruppen mit verschiedenen Berechtigungen) Zugriff auf welche Gruppen von Objekten und Funktionen haben.

Unter **Kommunikationsstrategie** wird für jede Benutzergruppe angegeben, in welcher grundsätzlichen Form der Dialog abläuft (Kommandosprache, Menüs, Maus/Cursor-Steuerung, durchgängige Hilfestellung, prinzipieller Bildschirmaufbau, spezielle Farbkodierung, Konfigurierbarkeit durch den Benutzer).

Der **Kommunikationsaufbau** wird durch das konkrete Layout aller Dialogmasken und Berichte spezifiziert. Er kann/sollte durch einen Oberflächenprototyp ersetzt werden.

Unter **Qualitätsziele** wird angegeben, welche der üblichen Qualitätsziele Vorrang haben: Effizienz, Zuverlässigkeit, Lebensdauer, Wartbarkeit, Portabilität.

Die **Entwicklungs-Umgebung** muß nicht mit der Produkt-Umgebung übereinstimmen. Für die Test- und Wartungsphasen müssen solche Unterschiede bekannt sein; vor allem, wenn der Kunde später die Wartung selbst übernehmen will.

Unter **Einführungsstrategie** sind die geplanten Maßnahmen zur Inbetriebnahme des Produkts zu beschreiben, z.B. Mitarbeiterschulung, Probetrieb in Teilen. Dies ist besonders wichtig bei Migration von einem Altsystem.

In **Anhängen** sind die im Pflichtenheft verwendeten Abkürzungen und Begriffe in einem Glossar festzuhalten, um eine einheitliche Terminologie zwischen Auftraggeber und Lieferanten herzustellen.

Quellen: Helmut Balzert : „Die Entwicklung von Software-Systemen“, Heide Balzert: Objektmod.
T. Pavlista: SE-Skript