

# Leitfaden zur Durchführung von Programmierpraktika

Teilweise entnommen aus:  
Helmut Balzert, Handbuch der Softwaretechnik, Spektrum 1996

## 1 Phasen eines Programmierprojektes

1. Planungsphase
2. Erstellung des Programmes
3. Dokumentation

Beim Bachelor gehören die Dokumentation und das Pflichtenheft in die Arbeit.

## 2 Planung

Bevor mit der eigentlichen Umsetzung eines Softwareprojektes (und damit einer Implementierung) begonnen werden kann, muß zunächst die Durchführbarkeit gezeigt werden.

Das Planen eines Projektes beinhaltet unter anderem folgende Aktivitäten:

- **Planungsphase**
  - Festlegen der Hauptanforderungen
    - \* Festlegen der Hauptfunktionen
    - \* Festlegen der Hauptdaten
    - \* Festlegen der wichtigsten Aspekte der Benutzerschnittstelle
    - \* Festlegen der wichtigsten Qualitätsmerkmale
  - Durchführbarkeit
    - \* Prüfen der funktionellen Durchführbarkeit (softwaretechnische Realisierbarkeit, Verfügbarkeit geeigneter Hardware)

- \* Prüfen der personellen Durchführbarkeit (ist man selber in der Lage den Anforderungen gerecht zu werden, evtl. Reduktion der Anforderungen im Vorfeld)

- **Pflichtenheft**

- Die Ergebnisse der oben genannten Punkte münden in einem Pflichtenheft.

- **Ziel**

- Die Tätigkeiten in der Planungsphase haben also das Ziel zu prüfen, wie ein Projekt entwickelt werden soll. Das Pflichtenheft bildet dann die Grundlage für das Projekt und damit für die Implementierung.

- **Ende der Planungsphase**

Mit Fertigstellung des Pflichtenheftes ist die Planungsphase beendet.

*Die Planungsphase endet, wenn der Betreuer das Pflichtenheft absegnet.*

### 3 Sinn und Zweck eines Pflichtenheftes

Das Pflichtenheft soll - grob gesprochen - den Umfang eines Softwareprojektes definieren.

Von der Seite des Programmierers aus gesehen schafft es klare Rahmenbedingungen, innerhalb derer er sich bewegen soll. Von der Seite der Betreuer aus gesehen dient es dazu (vor allem bei der Bachelor-Thesis), eine Bewertungsgrundlage zu definieren.

*Anhand des Pflichtenheftes soll das Projekt bewertet werden können.*

Da bei einer Bachelor-Thesis, das Pflichtenheft einen nicht unerheblichen Teil der Arbeit ausmacht, sollte hierauf besonders viel Wert gelegt werden.

Dieser Leitfaden soll als Hilfe bei der Erstellung von Pflichtenheften dienen, kein starres Schema darstellen, das in allen Punkten genau so verwendet werden soll.

## 4 Das Pflichtenheft

### 4.1 Definition

**Aufgabe:** Das Pflichtenheft enthält eine Zusammenfassung aller fachlichen Anforderungen, die das Softwareprojekt erfüllen muss.

**Inhalt:** Fachlicher Funktions-, Daten- und Leistungsumfang des Projektes. Beschreibung dessen **was** man tut und nicht **wie** man es tut. Die beschriebenen Anforderungen sollen realisierbar sein. Entwurfs- und Implementierungsentscheidungen sollen im Detail nicht vorweggenommen oder unnötig eingeschränkt werden.

*Was - nicht wie!*

**Form und Umfang:** Grobes Gliederungsschema mit festgelegten Inhalten, um Pflichtenhefte gut lesen zu können.

Die notwendigen Anforderungen müssen in ausreichender Detaillierung und ausreichendem Präzisionsgrad beschrieben werden. Der Funktions-, Daten- und Leistungsumfang muss aus Sicht der **Betreuer** auf hinreichendem Abstraktionsniveau vollständig beschrieben sein.

*So kurz wie möglich - so lang wie nötig!*

### 4.2 Gliederungsschema

1. **Einleitung:** Kurzer Überblick über die Aufgabe der das Projekt gewidmet ist.
2. **Zielbestimmung:** Hierunter fallen
  - (a) **Musskriterien:** Hier wird (grob) aufgeführt, welche Leistungen für das Programm unverzichtbar sind, damit es für den vorhergesehenen Zweck verwendet werden kann. Sie müssen auf jeden Fall erfüllt werden.
  - (b) **Abgrenzungskriterien:** Diese sind optional und sollen deutlich machen, welche Ziele innerhalb des Programms bzw. des Projektes nicht erreicht werden sollen.

3. **Anwendungsbereiche:** Da der geplante Einsatz des Programms wesentliche Auswirkungen auf die funktionale Mächtigkeit hat, werden in diesem Abschnitt die Anwendungsbereiche (z.B. Textverarbeitung) definiert.
4. **Funktionen:** Hier folgt die funktionale Beschreibung des Programmes aus Anwendersicht. Dieses Kapitel sollte in so viele Abschnitte gegliedert werden, wie das Projekt Funktionen oder Funktionsbereiche aufweist. Hier sollte darauf geachtet werden **was** und nicht **wie** etwas definiert wird. Was definiert wird sollte unabhängig sein von Bildschirm-Layout oder Tastaturbelegungen. Hier kann auch nochmal auf Wunschkriterien eingegangen werden. Um sich auf die einzelnen Funktionen beziehen zu können sollten diese eindeutig nummeriert werden.
5. **Daten:** Dies ist optional und dient der Beschreibung von langfristig zu speichernden Daten (z.B. in Dateien oder in Datenbanken).
6. **Qualitätsanforderungen:** Diese sind optional, und Beschreiben z.B. die Zuverlässigkeit, die Effizienz usw.
7. **Benutzungsoberfläche:** Falls vorhanden werden hier grundlegende Anforderungen an die Benutzeroberfläche definiert. Es sollten folgende Gesichtspunkte berücksichtigt werden:
  - (a) Bildschirmlayout
  - (b) Drucklayout
  - (c) Tastaturbelegung (z.B. Shortcuts o.ä.)
  - (d) Dialogstruktur (z.B. Assistentenartig, MDI, SDI)
8. **Programmumgebung**
  - (a) Unter **Software** wird angegeben, welche Software (Betriebssystem, Datenbanken, Frontend, Nicht-Standard-Bibliotheken usw.) auf einem Zielsystem zur Verfügung stehen muss.
  - (b) Unter **Hardware** werden die minimalen und maximalen Hardwareanforderungen (incl. Peripherie) für das Programm aufgeführt.

- (c) Unter **Orgware** wird aufgeführt, unter welchen organisatorischen Randbedingungen das Programm eingesetzt werden soll (sollen z.B. eMails versendet werden, braucht man einen LAN-Anschluss).
  - (d) Bei **Programm-Schnittstellen** werden die genutzten Schnittstellen zu anderen Programmen oder Bibliotheken (sofern nicht Standard) beschrieben (z.B. VBA-Schnittstellen).
9. **Entwicklungsumgebung:** Hier wird die Entwicklungsumgebung des Programmes beschrieben. Die Kategorien sind
- (a) **Software** (z.B. Compiler, Software-Werkzeuge)
  - (b) **Hardware**
  - (c) **Orgware**
10. **Sonstiges:** Hier kann alles stehen was noch nicht aufgeführt wurde, also z.B. Lizenzmodelle, Installationsspezifisches usw.

## 5 Dokumentation

Die Dokumentation besteht aus zwei Teilen und ist nach Implementierung zu erstellen. Der erste Teil umfasst die Quelltextdokumentation und dient dazu das Programm wartbar zu machen. Der zweite Teil ist eine Bedienungsanleitung, die entweder in Form eines Handbuches (*Buch* bezieht sich hier nicht auf den Umfang) oder einer Online-Dokumentation vorzulegen ist.