

- **Zu Grunde liegende Benutzerschnittstellen:** Wird ein Standard unterstützt? Werden damit nur bestimmte Applikationsklassen unterstützt? Wie viel Semantik und Funktionalität ist in den erzeugten Benutzerschnittstellen enthalten? Sind die zu Grunde liegenden Benutzerschnittstellen fest vorgegeben oder sind sie konfigurierbar?
- **Spezifikation der Benutzerschnittstellen:** Auf welcher Abstraktionsebene (z. B. als endlicher Automat, Hypertext, kontextfreie Grammatik etc.) werden sie spezifiziert? Welche Mittel (formale Sprachen; grafische Editoren, die nach dem WYSIWYG-Prinzip oder nach dem Prinzip der „direkten Manipulation“ arbeiten etc.) werden dazu eingesetzt? Wie werden statische Teile (Layouts), wie werden dynamische Teile (Abläufe, Zustandsübergänge) spezifiziert?
- **Verwendung der erzeugten Prototypen:** Können die Prototypen zu fertigen Applikationen erweitert werden? Kann aus ihnen lediglich ein Programmskelett erzeugt werden? Welche Informationen können wiederverwendet werden, und auf welche Weise (einfaches „Cut and Paste“ von Codeteilen; „höhere“ Verfahren, z. B. Manipulation von Benutzerschnittstellenkomponenten wie Dialogmuster, Dialogskelette, Bildschirm-layouts, etc.; Unterstützung durch Objektverwaltungssysteme)?
- **Unterstützung von Erweiterungen:** Mit welchem Aufwand sind Erweiterungen verbunden, die nur die Benutzerschnittstelle betreffen, und solche, welche die Funktionalität betreffen? Müssen dabei Compile-Link-Go-Zyklen durchlaufen werden oder liegt ein interpretierendes System zu Grunde? Kann die Benutzerschnittstelle zur Laufzeit verändert werden (z. B. kontextabhängige Menügestaltung, grafische Anzeige generierter Daten, benutzerabhängige Dialogstruktur)? Ist die Konsistenz der anfallenden Dateien und Dokumente gewahrt?
- **Art der funktionalen Erweiterungen:** Kann die Funktionalität der Prototypen schrittweise erweitert werden? Werden im Falle von Generatoren die Erweiterungen auf der Stufe der Spezifikation vorgenommen oder müssen sie im generierten Code implementiert werden? Wie ist bei Codeerzeugung die Qualität (Lesbarkeit, Strukturiertheit etc.) des generierten Codes? Ist die Codeerzeugung parametrisierbar (Wahl der Sprache, Wahl des Sprachdialekts, Formatierung)? Wie kann aus funktionalen Teilen des Prototyps auf die Benutzerschnittstelle zugegriffen werden? Können funktionale Erweiterungen, Ausnahmesituationen etc. simuliert werden? Ist es möglich, Prototypen herzustellen, bei denen manche Operationen ausgeführt, andere hingegen nur simuliert werden? Welche Schnittstelle besteht zu Datenbanksystemen?
- **Entwicklungsumgebung:** In welcher Entwicklungsumgebung kann das Prototypingwerkzeug eingesetzt werden und wie ist es in diese Umgebung integriert? Können Informationen zwischen den Prototypen und anderen Dokumenten ausgetauscht werden (z. B. über ein Datenlexikon)? Welche Softwarewerkzeuge (dynamische Debugger, Werkzeuge für die Versionsverwaltung, Testwerkzeuge, Profilingwerkzeuge etc.) und welche Programmiersprachen stehen zur Verfügung? Können mit dem Prototypingwerkzeug neben den Prototypen weitere Objekte (Dokumentationen, Online-Hilfdateien, Protokolle und Statistiken über Benutzerverhalten etc.) erzeugt werden, und wie können diese verwendet werden?
- **Qualität des Prototypingwerkzeugs:** Wie ist die Effizienz, die Zuverlässigkeit, die Komplexität und die Erlernbarkeit des Prototypingwerkzeugs?

- **Implementierung des Prototypingwerkzeugs:** Auf welchen Konzepten beruht die Implementierung? Ist das Prototypingwerkzeug erweiterbar?

8.1.5 Ergebnisse

8.1.5.1 Analysebericht

Die Ergebnisse der Analyse seitens des Auftragnehmers werden im *Analysebericht* dokumentiert und umfassen die Beschreibung des aus dem Istzustand zu entwickelnden Sollzustand in einer strukturierten, gegliederten Form. Dieser Bericht besteht im Allgemeinen aus der grafischen Darstellung der Problemstrukturierung sowie textuellen Erklärungen und Begründungen der Zerlegung.

Der Analysebericht dient zum Abgleich mit dem vom Auftraggeber erstellten Lastenheft und damit – neben den Planungsergebnissen, vgl. Kap. 8.2 – als Basis für die Erstellung eines *Angebots* bzw. für die *Einreichung* bei einer *Ausschreibung*.

8.1.5.2 Lastenheft

Wie z. B. in VDI-Richtlinien festgelegt, ist das *Lastenheft* (im Englischen oftmals *system requirements specification*; zu unterscheiden von der *software requirements specification* der IEEE [IEEE, 1994], welche um weitere IEEE-Dokumente, wie etwa *software development plan*, *quality assurance plan*, *statement of work*, erweitert werden muss) „die Zusammenstellung aller Anforderungen des Auftraggebers hinsichtlich Liefer- und Leistungsumfang“ [Verein Deutscher Ingenieure, 1991]. Im Lastenheft sind die Anforderungen aus Anwendersicht einschließlich aller Randbedingungen – quantifizierbar und prüfbar – zu beschreiben. Im Lastenheft wird definiert, WAS WOFÜR zu lösen ist.

Ziel des Lastenhefts ist es, die Projektaufgabe und das mit der Bearbeitung der Projektaufgabe verfolgte Ziel operational zu beschreiben, d. h. die Ziele des Projekts und dessen Bearbeitungsablauf müssen so verdeutlicht werden, dass sie von allen Projektmitarbeitern unmissverständlich nachvollzogen werden können. Projektziele sind dabei Sachziele („Was soll mit dem Projekt erreicht werden?“) und primär keine Planungsziele (Zeit-, Kostenziele etc.). Das Lastenheft wird aus diesen Gründen auch als *Aufgabenbeschreibung*, *Anforderungsdefinition* und – missverständlich – als *Systemspezifikation* bezeichnet.

Für das Lastenheft wird in [Verein Deutscher Ingenieure, 1991] folgende Gliederung vorgeschlagen, wobei projektabhängig Punkte ergänzt oder weggelassen werden können:

1. **Einführung in das Projekt, Projektziele:** Veranlassung, Zielpublikum, Zielsetzung, Projektumfeld, wesentliche Aufgaben, Eckdaten,
2. **Beschreibung des Istzustandes:** System, Prozess, Organisation, Datendarstellung und Mengengerüst (Ist),
3. **Beschreibung des Sollzustandes:** Kurzbeschreibung, Liste der Anforderungen (Analysegraph; Gliederung in Muss-, Soll- und Kann-Anforderungen), Ablaufbeschreibung, Beschränkungen, Abgrenzungskriterien (Nicht Enthaltenes), Datendarstellung und Mengengerüst (Soll), zukünftige Erweiterungen,

4. **Schnittstellen:** Übersicht: Systemebene/Protokolle, Mensch-Rechner, Rechner-Rechner, Rechner-externe Prozesse,
5. **Systemanforderungen:** Datenhaltung und -verarbeitung, Entwicklungs- und Zielhardware, Software, technische Merkmale und Leistungsdaten,
6. **Anforderungen an Inbetriebnahme und Einsatz:** Dokumentation, Testen, Inbetriebnahme, Abnahme, Schulung, Betrieb und Wartung,
7. **Qualitätsanforderungen:** Qualitätsmerkmale, Vorgehen bei der Qualitätssicherung, Nachweis der Produktqualität,
8. **Anforderungen an die Projektentwicklung:** Projektorganisation, Projektplanung (Meilensteine), Projektdurchführung (technische, monetäre, rechtliche Abmachungen),
9. **Anhang:** Begriffe und Definitionen, Abkürzungen, Nomenklatur, Gesetze, Normen und Richtlinien, Liste referenzierter Dokumente.

Das Lastenheft ist vom Auftraggeber vollständig und widerspruchsfrei zu erstellen. Es dient als Ausschreibungs-, Angebots- und/oder Vertragsgrundlage. In der Praxis wird jedoch vielfach der Auftragnehmer vom Auftraggeber mit der *Erstellung des Lastenhefts* beauftragt bzw. die Erstellung gemeinsam durchgeführt. Dies hat den gravierenden Nachteil, dass der Vergleich des Analyseberichts mit dem Lastenheft nur mehr begrenzt zur Überprüfung des Problemverständnisses herangezogen werden kann und die Gefahr besteht, das Projekt auf Basis einer ungenauen, unvollständigen oder gar schwerpunktmäßig falschen Analyse fortzusetzen. Das Lastenheft wird vom Auftraggeber geprüft und freigegeben.

Im Fall der Lastenhefterstellung durch den Auftragnehmer ist es entscheidend, sich in die Rolle und Sprachwelt des Auftraggebers zu versetzen und alle Formulierungen für den Auftraggeber verständlich und relevant zu gestalten. Diese Vorgangsweise unterstützt die WAS/WIE-Trennung, da im Lastenheft nur Anforderungen aus der Sicht des Auftraggebers („WAS“) zu beschreiben sind.

Checkliste für das Lastenheft

Diese Checkliste dient zur Vorprüfung des Lastenhefts durch den Auftragnehmer.

• Machbarkeit:

- Sind die Anforderungen vom Auftragnehmer (gemäß erstelltem Plan) realisierbar?
- Sind die Anforderungen im vom Auftraggeber geforderten Systemumfeld (Ziellplattform) realisierbar?
- Sind die Anforderungen mit ökonomisch vertretbarem Aufwand („vernünftig“) machbar? Gibt es ökonomisch unvernünftige Anforderungen (z. B. 100 % Fehlerfreiheit, Interaktivität bei allem und jedem etc.), selbst wenn sie vom Auftraggeber gefordert bzw. finanziert werden?

• Vollständigkeit:

- Fehlt eine in der Zielerforschung (Erhebungsbericht, bisherige Protokolle, Briefe etc.) erhobene Anforderung ohne explizite Angabe von Gründen?

- Fehlt eine vom Auftragnehmer als im Sinne des State-of-the-Art unverzichtbar eingestufte Anforderung?

• Minimalität:

- Ist jede im Lastenheft enthaltene Anforderung (in der Zielerforschung, persönlich vom Auftraggeber, durch den State-of-the-Art) auch verlangt?

• Konsistenz:

- Sind die Anforderungen untereinander widerspruchsfrei?

• Eindeutigkeit:

- Wird für jedes Attribut ein eindeutiger, festgelegter Ausdruck verwendet?
- Bei möglicher Mehrdeutigkeit oder vermutlicher Unklarheit für das Zielpublikum: Ist der Ausdruck im Glossar enthalten und dessen genaue Bedeutung festgelegt?

• Plausibilität:

- Ist jede Anforderung für den Auftragnehmer plausibel?

• Überprüfbarkeit:

- Ist die Realisierung der Anforderungen überprüfbar (qualitativ, quantitativ)?
- Ist die Realisierung der Anforderungen testbar?

• Modifizierbarkeit:

- Ist das Lastenheft übersichtlich gestaltet (Inhaltsverzeichnis vorhanden, adäquates Glossar enthalten, ggf. Index beigegeben)?
- Ist jede Anforderung separat angeführt?
- Sind die Anforderungen eindeutig gekennzeichnet (z. B. durch Nummerierung), um sie in späteren Phasen eindeutig referenzieren zu können?
- Wurde Redundanz vermieden, so weit dies sinnvoll möglich ist? Sind bei notwendigen Redundanzen entsprechende Querverweise angegeben?

8.1.5.3 Machbarkeitsstudie

Ist die Machbarkeit eines Projekts ungewiss, so wird manchmal zuerst nur eine Teilmenge des Lastenhefts ausgearbeitet, die sogenannte *Machbarkeitsstudie* (auch: *Durchführbarkeitsstudie*; *feasibility study*). Diese weist im Allgemeinen die gleiche Struktur auf wie ein Lastenheft, konzentriert sich jedoch in allen Gliederungspunkten darauf festzustellen, welche der Anforderungen technisch und wirtschaftlich durchführbar sind. Es wird entweder das gesamte Projekt auf Machbarkeit untersucht oder es wird versucht, ein machbares (und für den Auftraggeber sinnvolles) Teilprojekt zu definieren. Neben der Prüfung auf *technische Machbarkeit* und *wirtschaftliche Machbarkeit* ist oftmals auch – bei einer vorgegebenen Gruppe von Projektmitarbeitern – die *personelle Machbarkeit* zu überprüfen.

9.4.4.2 Grafische Generatoren zur Erstellung von Benutzerschnittstellen

Grafische Benutzerschnittstellen-Generatoren (*graphic user interface builder; GUI builder*) gewährleisten durch interaktive Eingabemöglichkeiten die rasche Änderbarkeit sowohl der Eigenschaften als auch des logischen Aufbaus der Oberflächenobjekte. Der Designer kann durch interaktive Manipulationen die Benutzerschnittstelle aus einer Reihe von Basisobjekten zusammenbauen. Das Aussehen der Schnittstelle kann durch Modifizieren verschiedener Attributwerte ebenfalls interaktiv unterstützt an die Benutzerwünsche angepasst werden. Aufgrund dieser Flexibilität ist der Oberflächendesigner in der Lage, durch Experimentieren das Aussehen und die Funktionalität der Schnittstelle optimal an die Bedürfnisse des Anwenders anzupassen.

Als zusätzliches Feature bieten einige dieser Systeme einen so genannten *Dialogmanager* an. Dieser Dialogmanager ist für die Kommunikation zwischen der grafischen Benutzerschnittstelle und der Applikation zuständig. Der Dialogmanager nimmt dem Oberflächendesigner einen großen Teil der Arbeit zum Verwalten der Benutzerschnittstellen-Komponenten ab. Da der Dialogmanager auch interaktiv betrieben werden kann, kann bereits in der Prototypingphase mit geringem Aufwand sehr viel Funktionalität in den Prototypen gepackt werden.

9.4.5 Werkzeuge des Compilerbaus

Die Motivation für die Verwendung *attributierter Grammatiken* für Designaufgaben ist die, dass in vielen Fällen der Programmablauf wesentlich durch den Strom der Eingabedaten gesteuert wird. Wenn man daher die Struktur dieser Daten durch eine Grammatik beschreibt und diese durch semantische Aktionen anreichert, kann man mittels Werkzeugen des Compilerbaus (z. B. Compiler-Compiler, vgl. [Dobler und Pirklbauer, 1990]) daraus ein Programm erstellen lassen, das dem gewünschten Ablauf entspricht. Diese Technik ist anwendbar, wenn im Wesentlichen nur ein Eingabestrom vorliegt und die Eingabeinformation genügend strukturiert ist [Pomberger, 1991].

Die Vorteile dieser Vorgehensweise sind die gut strukturierte Modularisierung des Problems, die äußerst präzise Problembeschreibung, aus der sich eine sehr exakte technische Dokumentation herleiten lässt, sowie die Möglichkeit der automatischen Programmgenerierung. Nachteile sind die Unverständlichkeit gegenüber vielen Auftraggebern (Compilerbaukenntnisse sind zum Verstehen attributierter Grammatiken notwendig) und die Beschränkung auf nur einen Eingabestrom.

Attributierte Grammatiken werden z. B. in [Coleman *et al.*, 1994] zur Beschreibung der zugelassenen Benutzerinteraktionen und deren Reihenfolge verwendet. Für weiter gehende Informationen zu attribuierten Grammatiken und Compilerbauwerkzeugen sei beispielsweise auf [Aho *et al.*, 1986] verwiesen.

9.5 Ergebnisse

9.5.1 Designbericht

Der *Designbericht* ist eine Erweiterung des Analyseberichts um die Designmodelle sowie gegebenenfalls erstellte Prototypen. Der Designbericht in seiner vollständigen Form wird normalerweise nicht dem Auftraggeber zur Verfügung gestellt, er dient der internen Dokumentation des Entwicklungs-Know-hows. Auszüge daraus sowie die Prototypen können jedoch dem Auftraggeber zur besseren Verdeutlichung der Inhalte des Pflichtenhefts zur Verfügung gestellt werden.

9.5.2 Pflichtenheft

Gemäß [Verein Deutscher Ingenieure, 1991] ist das *Pflichtenheft* (*system specification*) die Beschreibung der Realisierung aller Anforderungen des Lastenhefts. Das Pflichtenheft enthält das Lastenheft. Im Pflichtenheft werden die Anwendervorgaben detailliert und die Realisierungsanforderungen beschrieben. Im Pflichtenheft wird definiert, WIE und WOMIT die Anforderungen zu realisieren sind. Das Pflichtenheft beschreibt also sämtliche Aufgaben und Pflichten, die der Auftragnehmer im Rahmen des Projekts übernehmen muss. Es wird daher auch *Leistungsverzeichnis* genannt.

Da das Pflichtenheft eine Erweiterung des Lastenhefts darstellt, wird der Inhalt des Lastenhefts wie folgt im Pflichtenheft adaptiert:

1. **Einführung in das Projekt, Projektziele:** keine Änderungen zugelassen (bei Änderungen neues Projekt),
2. **Beschreibung des Istzustandes:** Ergänzungen bzw. Streichungen, Kommentare,
3. **Beschreibung des Sollzustandes:** Verfeinerung, Festlegung der zu erfüllenden und nicht zu erfüllenden Anforderungen,
4. **Schnittstellen:** Adaptierung, Detaillierung (z. B. Benutzerschnittstelle: Snapshots, Dokumentation, Interaktionsabläufe),
5. **Systemanforderungen:** Detaillierung der Systembedingungen (z. B. detaillierte Anforderungen an die Hardware, notwendige Systemsoftware wie etwa Compiler, Bibliotheken; Datenhaltungsmodell (Verwendung von Datenbanken, Datensicherung, Datenschutz und Datenarchivierung),

NEU: **Design:** Systemdesign, Komponentendesign,

6. **Anforderungen für Inbetriebnahme und Einsatz:** Verfeinerung, Adaptierung (z. B. Umfang und Detaillierungsgrad der Dokumentation),
7. **Qualitätsanforderungen:** evtl. Adaptierungen,
8. **Anforderungen an die Projektentwicklung:** Adaptierung sowie Verfeinerung des Zeitplans, Konkretisierung der Projektdurchführungsbestimmungen (z. B. Lieferung

der Abnahmedaten, Durchführung der Abnahme, Gewährleistung und Wartung, Rechte und Pflichten von Auftraggeber und Auftragnehmer – etwa bzgl. Quellcode; diese Bedingungen können aus dem Auftrag entnommen werden bzw. es kann auf den Auftrag verwiesen werden),

9. Anhang: Erweiterung.

Das Pflichtenheft wird in der Regel nach Auftragserteilung vom Auftragnehmer erstellt, falls erforderlich unter Mitwirkung des Auftraggebers. Zur Absicherung des Auftragnehmers ist im Pflichtenheft explizit anzuführen, was im Projektergebnis alles nicht enthalten sein wird (v. a. in Bezug auf Ergebnisse der Zielerforschung, vorangegangene Kontakte mit dem Auftraggeber, Lastenheft). Probleme treten jedoch häufig auch dann auf, wenn sich (Auftraggeber-)Spezifikationen nachträglich als zu ungenau herausstellen. Sich genau an die Spezifikation zu halten ist in diesem Fall nicht empfehlenswert, da der Auftraggeber sicher vielfach mit dem Ergebnis nicht zufrieden wäre. Besser ist es, die – ungenügende – Spezifikation als Minimalanforderung zu betrachten und gemeinsam mit dem Auftraggeber Verfeinerungen auszuarbeiten. Solche nachträgliche Adaptierungen bzw. Erweiterungen des Pflichtenhefts sind immer schriftlich festzuhalten und vom Auftraggeber und vom Auftragnehmer zu unterzeichnen; im Allgemeinen sind sie mit einer Vergrößerung des Projektumfangs verbunden.

Der Auftragnehmer prüft bei der Erstellung des Pflichtenhefts die Widerspruchsfreiheit und Realisierbarkeit der im Lastenheft genannten Anforderungen (so er nicht bereits an der Erstellung des Lastenhefts beteiligt war). Das Pflichtenheft bedarf der Genehmigung durch den Auftraggeber („Abnahme“). Nach der Abnahme wird das Pflichtenheft die verbindliche Vereinbarung für die Realisierung und Abwicklung des Projekts für Auftraggeber und Auftragnehmer.

Checkliste für das Pflichtenheft

Diese Checkliste dient zur Vorprüfung des Pflichtenhefts durch den Auftragnehmer. Ein erneutes Durchgehen der Checks des Lastenhefts für das Pflichtenheft ist sehr empfehlenswert.

• Kompatibilität:

- Ist das Design mit dem Lastenheft kompatibel?
- Sind Anforderungsänderungen zum Lastenheft explizit im Pflichtenheft dokumentiert?

• Technische Qualität:

- Wird das Design vom Auftragnehmer als gut beurteilt?
- Sind bei mit „nicht gut“ beurteilten Designbereichen die Gründe bekannt, akzeptiert und dokumentiert?

• Realisierbarkeit:

- Ist das Design in dem im Pflichtenheft festgelegten Umfeld vom Auftragnehmer in ein Produkt umsetzbar?
- Wird die Zeitplanung als realistisch eingestuft?

9.5.3 Benutzerdokumentation

Um die Designergebnisse bei der Implementierung leicht zugänglich zu haben, ist es sehr sinnvoll, die Benutzerdokumentation (bzw. die erste Version davon) bereits während der Designphase zu erstellen. Darüber hinaus können durch ein Review der Benutzerdokumentation mit dem Auftraggeber Designfehler bereits vor der Implementierung erkannt werden.

Die Benutzerdokumentation umfasst folgende Komponenten:

1. **Allgemeine Produktbeschreibung:** Die allgemeine Produktbeschreibung informiert über den Anwendungsbereich des Produkts und über dessen Funktionalität bzw. nicht abgedeckte Funktionalität. Sie enthält Angaben über benötigte Hard- und Software-Ressourcen, organisatorische und infrastrukturelle Voraussetzungen sowie ggf. einfache Anwendungsbeispiele.
Ein Interessent soll den Zweck des Produkts verstehen und feststellen können, wie er das Produkt am besten nutzen kann. Einzelheiten zum Produkt sind nicht erwünscht, ein Überblick über die gesamte verfügbare Benutzerdokumentation soll jedoch gegeben werden.
2. **Installationsanleitung:** Die Installationsanleitung beschreibt, was an Hardware und Software zur Installation notwendig ist sowie den Vorgang der Installation selbst. Sie umfasst auch eine Liste aller zum Lieferumfang gehörigen Dokumente und Komponenten.
3. **Anweisungen zur Inbetriebnahme:** Diese umfassen den Aufruf des Systems, eine Liste dabei möglicher Fehler mit deren Behandlung, den Zugang zu Hilfefunktionen sowie eventuell ein Beispiel.
4. **Kurzeinführung (Tutorial):** Das Ziel der Kurzeinführung ist es, einem Neuling rasch den Betrieb des Systems mit seinen wesentlichsten Funktionalitäten zu ermöglichen. Daher ist dieses Dokument am besten so aufgebaut, dass der Inhalt Schritt für Schritt am System nachvollzogen werden kann. Ein Tutorial kann auch (oder ergänzend) in Online-Form verfügbar sein.
5. **Benutzerhandbuch:** Das Benutzerhandbuch gibt eine Einführung in das Programm für den täglichen Gebrauch. Die Beschreibung muss dabei so erfolgen, dass nur über die für den Benutzer erkennbare Funktionalität (via Benutzerschnittstelle) gesprochen wird, die interne Funktionalität bzw. interne Abläufe werden nicht angesprochen.

Das Benutzerhandbuch enthält die Beschreibung der Systemfunktionalität in leicht verständlicher Form sowie die jeweils notwendigen Eingaben und erzielbaren Ausgaben. Alle Ein- und Ausgabedaten werden erläutert und begründet, am besten anhand von im typischen Gebrauch vorkommenden Beispielen. Eine Liste aller Fehlermeldungen soll das Ermitteln von Fehlerursachen und Maßnahmen zu deren Behebung ermöglichen. Ein Kapitel soll sich mit den Möglichkeiten zur individuellen Anpassung des Systems an den Benutzer beschäftigen.

Das Benutzerhandbuch soll so geschrieben sein, dass es sequenziell gelesen werden kann, wobei die Sprache für den Anwender verständlich sein muss. Dies ist besonders wichtig