

Kurzanleitung Projektdokumentation

Diese Kurzanleitung zum Thema „Projektdokumentation“ gibt einen Überblick über Normen zur Projektdokumentation und liefert im zweiten Teil eine Variante der Dokumentation wie sie für das Softwaretechnik-Praktikum sinnvoll sein kann.

1 Normen zur Projektdokumentation

Die beiden grundlegenden Dokumente der Projektdokumentation sind das Lasten- und das Pflichtenheft. Sie sind durch Normen beschrieben. Die beiden folgenden Abschnitte zeigen für beide Dokumente Gliederungsschemen auf Basis dieser Normen.

1.1 Lastenheft

Ein Lastenheft fasst die Anforderungen an die Leistungen und Lieferungen des Auftragsnehmers zusammen. Es notiert die funktionalen und qualitativen Anforderungen in einer überprüfbareren Form, definiert die vorliegende Aufgabe und für welchen Zweck sie zu lösen ist.

Für Lastenhefte existieren Normen, z.B. DIN 69905, diese werden von Unternehmen unterschiedlich umgesetzt. Ein Beispiel für ein Gliederungsschema ist aus [1, Seiten 62 u. 63] entnommen:

1. **Zielbestimmung:** Welches Ziel soll mit dem Produkt erreicht werden?
2. **Produkteinsatz:** Anwendungsbereiche und Zielgruppen
3. **Produktübersicht:** Überblick über die Produktumgebung, textuell und/oder graphisch
4. **Produktfunktionen:** Kernfunktionen des Systems aus Anwendersicht
5. **Produktdaten:** Wichtigste persistente Daten, Informationen im System
6. **Produktleistungen:** Leistungsanforderungen, z.B. Performance, Mengengerüst, Genauigkeit etc.
7. **Qualitätsanforderungen:** Weitere Qualitätsmerkmale, z.B. Erweiterbarkeit etc.
8. **Ergänzungen:** Spezielle Anforderungen, etwa an die Benutzeroberfläche

1.2 Pflichtenheft

Das Pflichtenheft beinhaltet die Realisierungsvorgaben zur Umsetzung des Lastenhefts, also die Umsetzung der Anforderungen in eine konkrete Lösung. Es definiert, wie und wodurch die Anforderungen zu erfüllen sind.

Die IEEE sieht in Standard 830 für Software Requirement Specifications, also für ein Pflichtenheft (u.a. zu finden in [3, Seite 224 ff.]) folgende Gliederung vor:

1. **Ausgangssituation und Zielsetzung** Wie sieht die derzeitige Situation aus? Wie lautet das Ziel?
2. **Systemeinsatz und Systemumgebung:** Wie sind die äußeren Bedingungen?
3. **Benutzerschnittstellen:** Welche Benutzerschnittstellen sind geplant?
4. **Funktionale Anforderungen** Inhaltliche Anforderungen
5. **Nichtfunktionale Anforderungen** Anforderungen wie Stabilität, Erweiterbarkeit, etc.
6. **Fehlerverhalten:** Grundsätzliches zum Umgang mit Fehlern
7. **Dokumentationsanforderungen:** Welche Dokumente sind im weiteren Verlauf nötig?
8. **Abnahmekriterien:** Durch welche Kriterien ist definiert, wann das Produkt fertig ist?
9. **Glossar und Index:** Glossar mit den wichtigen Begriffen

2 Rezept für das SWT-Praktikum

Die Normen werden sehr unterschiedlich angewandt. Die konkrete Anwendung ist vom Auftraggeber und dem Kontext abhängig.

Das Folgende ist eine „kreative Anwendung“ der Normen. Es stellt eine Möglichkeit dar, die Dokumentation für das Softwaretechnik-Praktikums zu erstellen und hat zum Ziel diejenigen Dokumente bereitzustellen, die für das Entwickeln und Verstehen nötig sind. Es ist als ein Vorschlag anzusehen, der an die Gegebenheiten der Softwaretechnik-Praktikums angepasst werden muss.

2.1 Exposé

Das Exposé dient dazu, den Zweck und Nutzen des Projektes darzustellen. Es sollte Aussagen darüber enthalten welches Problem gelöst wird, wofür und warum es dieses Projekt gibt. Auch die wichtigsten Anforderungen können an dieser Stelle notiert werden und welche Infrastruktur benötigt wird. Der Umfang sollte ca. 1,5 Seiten betragen.

Tipp: Häufig sind 80% des Exposés in der Aufgabenstellung zu finden.

2.2 Spezifikation

Die Spezifikation beschreibt die externe Sicht des Programms. Sie drückt gegenüber dem Verwender aus, wie es verwendet wird. Sie sollte eine detaillierte Beschreibung der Funktionalität enthalten, die z.B. durch Anwendungsfälle oder Verwendungsszenarien erläutert werden.

Des weiteren sollten die Qualitätsanforderungen, die nicht funktionalen Anforderungen, definiert werden. Dazu gehören Parameter wie Zuverlässigkeit, Wartbarkeit und die Umgebungsbedingungen.

Ausserdem sind Aussagen über die Verwendung also die Schnittstelle(n) zu treffen: Handelt es sich um ein GUI-Programm, ist eine Skizze des Interfaces sinnvoll, untermalt durch Abbildungen und Erläuterung des Aufbaus der Navigation. Wenn das Programm über die Kommandozeile bedient wird, ist eine „man-Page“-artige Dokumentation zweckmäßig.

2.3 Entwurf

Der Entwurf dient der Darstellung der internen Struktur. In ihm sollten im besonderen die Entscheidungen, die hinter der Architektur stecken, erläutert werden. Meist sind es Entwurfsentscheidungen, die später nur sehr schwer geändert werden können.

Eine mögliche Einteilung wäre die Komponenten auf zwei Ebenen zu unterteilen: einerseits die größeren Komponenten andererseits die kleineren Einheiten / Klassen. Es ist nicht notwendig, jede Klasse und jedes Details zu beschreiben.

Ziel ist es das „Spannende“ herauszugreifen, je nach Problemstellung kann das ganz verschieden sein, z.B.: der Algorithmus, der dahinter steckt, die Datenstruktur oder auch die Besonderheiten der Architektur. Um es nochmal zu unterstreichen, es geht um die Kernidee nicht um jedes Detail.

2.4 API-Dokumentation

Die API-Dokumentation dient dem Programmierer als Online-Referenz. Sie sollte aus dem Quellcode mit Hilfe von Javadoc generiert werden (siehe [2, Kapitel 3.2]). Nur auf diese Weise ist sie immer aktuell zu halten und hat einen Nutzen im täglichen Gebrauch.

2.5 Benutzerdokumentation

Die Benutzerdokumentation ist für den Endanwender und dient ihm als Hilfestellung für die Verwendung des Programms. Sie ist, wenn die Spezifikation sehr gut war, bereits in ihr enthalten. Denn inhaltlich werden Elemente der Spezifikation wiedergegeben, aber in einer Form, die für den Entanwender geeignet ist. Zusätzlich sind Hinweise zur Installation sinnvoll. Als Format bietet sich je nach Anwenderschnittstellenart eine Online-Hilfe-Dokument oder eine rein textuelle Anleitung an.

Literatur

- [1] Helmut Balzert. *Lehrbuch der Software-Technik - 1.Band*. Spektrum - Akademischer Verlag, 2000.
- [2] B. Franzen, B. Iglar, N. Krümmel, T. Letschert & B. Renz. *Java-Codierrichtlinien für den Fachbereich MNI*. FH-Gießen-Friedberg, 2007.
- [3] Hans van Vliet. *Software Engineering*. Wiley, 2003.

Autor: BURKHARDT RENZ, NADJA KRÜMMEL, Projektgruppe Softwaretechnik.