



Systemdenken und Gestaltungsmethodik

Dokumentation

Prof. Dr.-Ing. Stefan Brunthaler
TFH Wildau 2007ff
Master Telematik

Einige Grund-Tatsachen...

- Entwickler wollen nicht dokumentieren
- Anwender wollen nicht lesen
- Entscheider wollen nicht mitdenken
- Projektleiter wollen rechtzeitig fertig werden und das Budget (kl)einhalten
- **Ergo: Keine Doku erforderlich?**

Mehr Grundtatsachen

- Entwickler können nicht “schreiben”
- Anwender haben keine Zeit
- Entscheider haben auch keine Zeit
- Projektleiter beherrschen nur Telefon, Email, Powerpoint und MS Project
- **Ergo: Keine Doku möglich?**

Wozu überhaupt Doku?

- **Anwender** sollen das Produkt bedienen
- **Entwickler** sollen das Produkt pflegen
 - ... und ggf. auch weiterentwickeln
- **Entscheider** sollen den Nutzen erkennen
- **Projektleiter** sollen den Stand verstehen
- **Ergo: Doku ist für alle da!**

Zielgruppen

- Projektleiter, Planer,
- Designer, Entwickler, Programmierer,
- Anwender,
- Ausbilder, Administratoren, Support,
- Interessenten, Entscheider, Verkäufer,
- Fachpresse, Bestandskunden, ...

Dokumentationstypen

- Prozess-Dokumentation (kurz):
 - Projektverlauf,
 - Design-Entscheidungen,
 - Entwicklungs-Verfahren und Regeln,
 - Support- und Pflege-Prozeduren.
 - Produkt-Dokumentation
 - Siehe nächste Folie →
- Quelle: Sommerville 1992 (!)

Produkt-Dokumentation

- System-Dokumentation:
 - Requirements, Lasten/Pflichtenheft
 - Architektur und Design
 - Technische Beschreibungen
- Anwender-Dokumentation:
 - Benutzungs-Anleitungen und Hilfen
- Werbe- und Marketing-Dokumente

Architektur und Design

- SysML und UML Dokumente entstehen in der Regel „automatisch“
- Verbale Beschreibungen („Vision“)
- Vereinfachte Grafiken für ein breiteres Publikum ohne Informatik-Hintergrund
- Hardware-Komponenten nicht vergessen: Funktionsstrukturen!

Technische Beschreibungen

- Konstruktionszeichnungen, Blueprints (HW)
- Spezifikation und Design-Beschreibung jeder verwendeten Komponente (HW, SW)
- Beschreibung der Architektur jedes einzelnen Programmes / Moduls (SW)
- Gültigkeit und Korrektheit des Systems mit Bezug auf die Anforderungen („Scope“)
- Kommentierter Source Code →

Kommentare im Code

- Ziel ist darzustellen, was der Code tun soll und warum es so und nicht anders implementiert wurde.
- Bezug zu den Anforderungen herstellen.
- Syntax der Sprache muss bekannt sein!
- Sinnfälligkeit „eleganter“ Codierungen, ...
 - ... wenn der Kommentar 10x so lang wird?

Hinweise zur Systemdoku

- Zielsetzung ist der **Werterhalt** des Produktes: Langfristige Nutzbarkeit, Pflegbarkeit und Erweiterbarkeit!
- Die Doku muss für einen **Neuling** verständlich sein und **schnell** zu **Produktivität** führen!
- Hauptproblem: **Nebeneffekte** und Querbeziehungen in grossen Systemen!
- **Quelltext-Analyse ist was für Theologen!**

Anwender-Doku (1)

- Funktionsbeschreibung
- Getting started oder „beim ersten Mal...“
- Referenz:
 - Listet die Funktionen auf, wie man sie erreicht und was sie können
- Bedienerhandbuch:
 - erklärt an typischen Anwendungsfällen, wie man Aufgaben löst

Anwender-Doku (2)

- Ad-Hoc Online-Hilfen:
 - Help-Link pro Programm / Formular
 - Hilfetexte auf Feldebene
 - Support-Webseite des Anbieters
- Zur Vertiefung / Community:
 - Anwender-Wiki
 - Foren- oder News-Server
 - Mailing-Listen

Sie müssen nur den Nippel...

Auszug aus der Bedienungsanleitung des Autoradios Blaupunkt DAB54:

Seite 63:

Seite 3:



Was ist der Betrachtungswinkel?

Betrachtungswinkel einstellen

Sie können den Betrachtungswinkel einstellen um das Gerät an die Einbaulage in Ihrem Fahrzeug anzupassen.

- Drücken Sie die Taste **MENU** (9).
- Drücken Sie die Taste **↵** oder **↶** (7) so oft, bis „DISPLAY MENU“ im Display angezeigt wird.
- Drücken Sie die Taste **OK** (10).
- Drücken Sie die Taste **↶** oder **↵** (7) so oft, bis „ANGLE“ mit der aktuellen Einstellung im Display angezeigt wird.
- Drücken Sie die **<** oder **>**-Taste (7), um den Betrachtungswinkel einzustellen.

Wenn der Einstellvorgang abgeschlossen ist,

- drücken Sie die Taste **MENU** (9).

Hinweise zur User-Doku

- Nichts voraussetzen, Erklärungen bieten!
- Unterschiedliche Qualifikationen und Medien-Erfahrungen berücksichtigen!
- Einfache Sprache, gut lesbare Schrift!
- Inhaltsverzeichnis, Stichwortverzeichnis!
- Bei digitaler Doku auf Usability achten!
- Interaktion ermöglichen!

Vertrieb und Marketing

- Produkt-Flyer (Image + Technik)
- Werbliche Funktionsbeschreibung
- Preislisten (mit Bezug auf die FB?)
- Webseite des Anbieters, Konfigurator
- Unique Selling Points Checkliste
- Vergleich mit Mitbewerbs-Produkten

Redundanz-Problem

- Zu einer Systemeigenschaft gibt es Inhalte in verschiedenen Dokumenten
- Jeweils wird ein etwas anderer Aspekt erklärt, aber bestimmte Teile sind gleich
- Dies bedeutet erheblichen Aufwand bei der Aktualisierung und Pflege

Wie kann man Abhilfe schaffen?

Motivations-Problem

- Designer: Nimm das UML-Diagramm, da steht doch alles drin!
- Entwickler: Kannst Du etwa keinen Quellcode lesen?
- Oder auch: Bevor ist das beschrieben habe, habe ich es schon dreimal (selber) programmiert.

Warum ist das so? Abhilfe?

Mein persönliches Fazit (1)

- Das größte Hindernis sind die fehlenden Standards bzgl. Tools und Strukturen
- Gleich danach kommt die Zugangs-Barriere: Wo stand das nochmal, und wie komme ich da dran?
- Auch problematisch ist die Pflege redundanter „Fragmente“.

Fazit (2)

- Einfach und offen, aber kontrollierbar:
 - Wiki mit Redakteur für User
 - Wiki mit Redakteur und Bewertungs-Tool für Entwickler
 - Cross-Verlinkung für redundante Inhalte
- Zur Quellcode-Kommentierung vorhandene Tools verwenden, etwa <http://kelp.sourceforge.net/> (ohne Gewähr)

Quellen

- Wikipedia via Google, Suchbegriff „software documentation“
- Sommerville: Software Engineering (1992)
- Grünwied: Software-Dokumentation (2007)
- Scheibl: Wie dokumentiere ich ein DV-Projekt? (1985)
- Interne Standards meiner Firma
- Das Internet natürlich!