

SoDoM: Software-Dokumentation mit Mustern – eine Projektskizze*

Klaus Quibeldey-Cirkel
Universität Siegen, FB Elektrotechnik und Informatik
D-57068 Siegen
quibeldey@ti.et-inf.uni-siegen.de

Das Thema Softwaredokumentation ist *unbedeutend* – das suggerieren zumindest die Standardwerke der modernen Entwurfslehre; sie erwähnen das Thema nur beiläufig. Folglich findet die Technische Dokumentation wenig Aufmerksamkeit in der Ausbildung. Auch moderne *Upper-CASE*-Werkzeuge, wie Rational Rose® oder objectiF®, bieten kaum Unterstützung: zum einen fehlt die Beschreibung einer Methode und ihre Integration in ein Vorgehensmodell, zum anderen die werkzeugtechnische Infrastruktur.

Zusammenfassend läßt sich nur das Fehlen einer Dokumentationskultur konstatieren, und es ist verständlich, wenn der Software-Ingenieur selten Antrieb verspürt, seine Arbeit mitzuteilen. Auch gibt das Projektmanagement kurzfristigen Aufgaben den Vorrang und übersieht dabei die langfristige Entwicklung: Das über die Jahre aufgebaute *IP*-Firmenkapital (*Intellectual Property*) verbleibt in den Köpfen der Entwickler und läuft Gefahr, mit ihnen zu gehen (*Brain-Drain*). Die Einarbeitung neuer Mitarbeiter wird mit der Projektgröße immer aufwendiger; auch hier lautet die *Crux*: Die Folgen kurzfristigen Handelns wirken sich langfristig aus.

Fragt man bei Unternehmen der Softwarebranche nach, ergibt sich ein differenzierteres Bild: Der Status quo ist unbefriedigend, der Dokumentation wird eine wachsende Bedeutung zugeschrieben, die Probleme sind weithin ungelöst. Gerade in Zeiten der Abkehr vom Wasserfallmodell, der Hinwendung zum iterativen Vorgehensmodell, des Einsatzes objektorientierter Entwurfsmethoden und der Wende vom Handentwurf zur komponentenbasierten Produktion bedarf es neuer Antworten auf alte Fragen. *Denn*: Mit der Einführung der Objekt-Technik ist die Erwartung an mehr Produktivität gekoppelt – der lang gehegte Wunsch, aus einer Bausteinsammlung die passenden auszuwählen und damit neue Software einfacher, billiger und schneller zu konstruieren. *Aber*: Die effektive Wiederverwendung eines Bausteins setzt dessen methodische Dokumentation voraus, siehe [Zendler et al., 1995].

Neues Denken

Entwurfsmuster zählen zu den meistdiskutierten Themen unter Informatikern [Gamma et al., 1995]. Dabei sind Muster nur insoweit etwas Neues, als sie vorhandenes Wissen strukturiert vermitteln. Warum also die Euphorie? Wir meinen, sie ist Ausdruck eines *Defizits* heutiger Softwareprojekte. Vergleicht man die typische Entwurfsituation eines Software-Ingenieurs mit der eines Architekten, so ist der Architekt konfrontiert mit Verordnungen und Normen, die ihn zwar einerseits reglementieren, andererseits aber seine Arbeitswelt vorstrukturieren, siehe zum Beispiel die VOB, Verdingungsordnung für Bauleistungen [VOB, 1979]. Der Software-Ingenieur dagegen arbeitet weitgehendst künstlerisch: das Arbeitsergebnis hängt von seiner Kreativität ab; eine Qualitätskontrolle ist kaum möglich.

Die Mustergemeinschaft hat sich zum Ziel gesetzt, auch den Prozeß des softwaretechnischen Entwerfens im positiven Sinne zu reglementieren: Entwurfswissen wird als Handbuch aufgeschrieben, analog zur VOB des Architekten. Der Jungingenieur kann sich am Erfahrungswissen des Experten orientieren und so auf einem höheren Niveau einsteigen. Die Strukturierung durch Muster erleichtert das Verständnis komplexer Entwürfe; an dieser Stelle setzt das Projekt an.

* Kurzfassung der Projektbeschreibung aus [Quibeldey-Cirkel, 1999, S. 153-161]

Projektziel

Bislang werden Entwurfsmuster entweder *deskriptiv* zur Weitergabe von Erfahrungswissen eingesetzt oder *generativ* zur Produktion objektorientierter Software. Wir verwenden die Musterform zur Strukturierung der Dokumente und den Musterinhalt als Vorbild für das Dokumentieren. Der Effekt ist eine wesentlich einfachere Integration der Dokumentation in die musterbasierte Entwicklung, die Qualität der Dokumente steigt, die Entwicklung läßt sich zurückverfolgen.

Thesen

- Ein musterbasierter Entwurf kann mit konventionellen Techniken auch musterbasiert dokumentiert werden. Entwurfsmuster sind neben ihrer anerkannten Eignung für die Softwarekonstruktion und für den Erfahrungstransfer auch ein *strukturebendes* und *vorbildhaftes* Vehikel für das Dokumentieren [Quibeldey-Cirke, 1999].
- Das Verständnis komplexer Entwürfe wird durch die musterbasierte Dokumentation wesentlich erleichtert, was den Aufwand für die Einarbeitung neuer Mitarbeiter reduziert [Odenthal & Quibeldey-Cirke, 1999].
- Softwarekomponenten, dokumentiert nach dem Musteransatz, sind leichter wiederverwendbar, da ein Bezug zum Muster besteht und dieses die konkrete Anwendung verallgemeinert.
- Das Konzept eignet sich auch zur Nachdokumentation alter Entwürfe, wenn sie hinreichend strukturiert sind. Erfahrungen liegen vor: [Meusel et al., 1997; Blachnik, 1997].
- Die Validierung der Methode in Pilotprojekten [Odenthal & Quibeldey-Cirke, 1996, 1997] hat gezeigt, daß sich die Dokumentation in die Software-Entwicklung und Werkzeugumgebung sinnvoll integrieren läßt. Das Schreiben wartungsfreundlicher Softwaredokumente wird *rationalisierbar*.

In angewandter Forschung wird derzeit eine Software-Entwicklungsumgebung (SEU) aus *markt-gängigen* Produkten konfiguriert; sie organisiert die Produktion von Softwarebausteinen, deren Verwaltung und den Zugriff darauf. Die SEU-Infrastruktur unterstützt die musterbasierte Vorgehensweise und läßt sich an den Standard-Schnittstellen leicht erweitern.

Hypertext als Basistechnik

Die Erweiterung der konventionellen SEU besteht darin, die Entwicklungsdaten, das sind Texte und Grafiken aller Art, die im Verlauf eines Softwareprojekts anfallen, durch ein *Informationssystem* zu präsentieren (siehe Bild 1) – in Form eines heterogenen Hypertextes (Bild 2). Der Anwender erschließt den Datenbestand mit formularbasierten Suchanfragen und durch Navigieren. Das System beantwortet die Fragen mit jeweils dynamisch zusammengestellten HTML-Seiten. Navigiert wird nicht nur durch das Folgen von Hyperlinks, sondern auch mittels moderner Techniken: *Fish-eye View* [Nielsen, 1993], dynamischer Verzeichnisbäume zur Darstellung benachbarter Knoten, *Tree View*, und *Guided Tour*. Die Basistechnik Hypertext ist der zweckmäßige Ansatz, um objektorientierte Software zu dokumentieren. Durch Links können Beziehungen unterschiedlicher Granularität zwischen den Dokumenten konstruiert werden, beispielsweise zwischen einem Textdokument und einem Grafikelement im Klassendiagramm.

Über den aktuellen Stand des Projekts informiert die Website:

 http://www.ti.et-inf.uni-siegen.de/Entwurfsmuster/ArtikelImport/index_Software_Dokumentation.htm

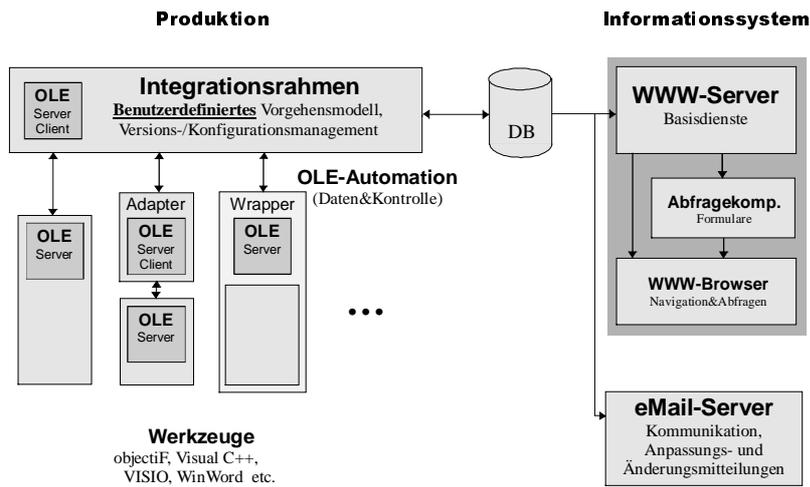


Bild 1:
Infrastruktur einer
Software-Entwicklungsumgebung,
erweitert für die Dokumentation

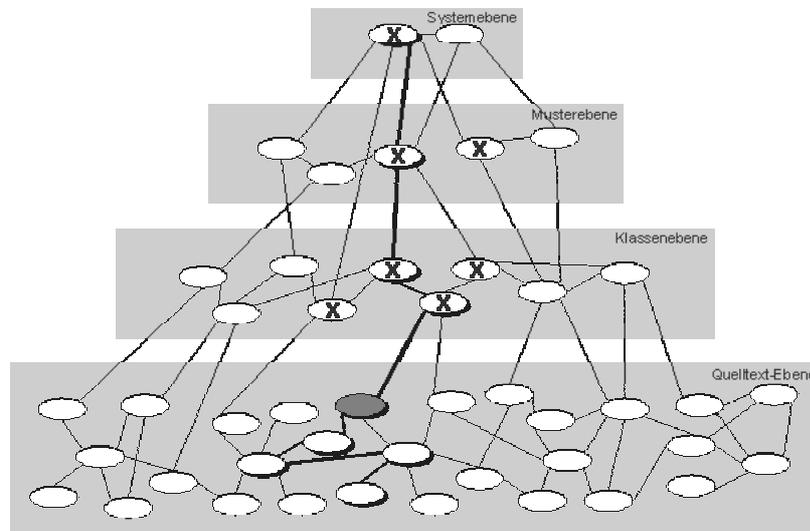


Bild 2:
Hypertext-Gesamtstruktur
mit einer *Guided Tour* (dicke Linie),
den bisher besuchten Knoten (x)
und dem aktuellen Knoten (graues Oval)

[Blachnik, 1997] Blachnik, Markus: Konzeption und Implementierung einer HTML-basierten Werkzeugumgebung zur Dokumentation mit Entwurfsmustern. Siegen: Universität Siegen, 1997, Diplomarbeit.

[Gamma et al., 1995] Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John: Design Patterns: Elements of Reusable Object-Oriented Software. Reading MA u. a.: Addison Wesley, 1995.

[Meusel et al., 1997] Meusel, Matthias; Czarnecki, Krzysztof; Köpf, Wolfgang: A Model for Structuring User Documentation of Object-Oriented Frameworks Using Patterns and Hypertext. In: Proc. of ECOOP 97. Berlin u. a.: Springer-Verlag, Reihe: Lecture Notes in Computer Science, Bd. 1241, 1997. S. 496-510.

[Nielsen, 1993] Nielsen, J.: Hypertext and Hypermedia. Cambridge: Academic Press Inc., 1993.

[Odenthal & Quibeldey-Cirke, 1996] Odenthal, Georg; Quibeldey-Cirke, Klaus: Mustergestützter Entwurf und Dokumentation mit Entwurfsmustern: Erfahrungen aus einem SAP-Projekt. In: Pötschke, Dieter; Weber, Mathias (Hrsg.): Anwendungen für Kommunikations-Highways: Perspektiven in den neuen Bundesländern. Konferenzband zur INFO 96. Heidelberg: R. v. Decker's Verlag, 1997. S. 326-332.

[Odenthal & Quibeldey-Cirke, 1997] Odenthal, Georg; Quibeldey-Cirke, Klaus: Using Patterns for Design and Documentation. In: Proc. of ECOOP 97. Berlin u. a.: Springer-Verlag, Reihe: Lecture Notes in Computer Science, Bd. 1241, 1997. S. 511-529.

[Odenthal & Quibeldey-Cirke, 1999] Odenthal, Georg; Quibeldey-Cirke, Klaus: Workshop Report on Pattern-Aided Software Documentation. In: Coldewey, Jens; Dyson, Paul (Hrsg.): Proceedings of the 3rd European Conference on Pattern Languages of Programming and Computing, 1998. Konstanz: Universitäts-Verlag Konstanz, 1999.

[Quibeldey-Cirke, 1999] Quibeldey-Cirke, Klaus: Entwurfsmuster: Design Patterns in der objektorientierten Softwaretechnik. Heidelberg u. a.: Springer-Verlag, 1999.

[VOB, 1979] Verdingungsordnung für Bauleistungen. VOB. Im Auftr. d. Dt. Verdingungsausschusses für Bauleistungen. Hg. vom DIN, Dt. Inst. für Normung e.V., Ausgabe 1979. Berlin, Köln: Beuth-Verlag, 1979.

[Zendler et al., 1995] Zendler, A.; Gasting, S.; Hagenmüller, R.: Vergleichende Analyse von Werkzeugen zum Aufbau und Einsatz von Bibliotheken für wiederverwendbare Software-Dokumente. München: Forschungsinstitut für Angewandte Software-Technologie, FAST e.V., Februar 1995.