

Kai Jan Oey

Dokumentation in der Softwareentwicklung - Effizienter Einsatz von Entwicklerdokumentation

Diplomarbeit

BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren



Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Impressum:

Copyright © 2002 GRIN Verlag
ISBN: 9783638252966

Dieses Buch bei GRIN:

<https://www.grin.com/document/21755>

Kai Jan Oey

Dokumentation in der Softwareentwicklung - Effizienter Einsatz von Entwicklerdokumentation

GRIN - Your knowledge has value

Der GRIN Verlag publiziert seit 1998 wissenschaftliche Arbeiten von Studenten, Hochschullehrern und anderen Akademikern als eBook und gedrucktes Buch. Die Verlagswebsite www.grin.com ist die ideale Plattform zur Veröffentlichung von Hausarbeiten, Abschlussarbeiten, wissenschaftlichen Aufsätzen, Dissertationen und Fachbüchern.

Besuchen Sie uns im Internet:

<http://www.grin.com/>

<http://www.facebook.com/grincom>

http://www.twitter.com/grin_com

Dokumentation in der Softwareentwicklung. Effizienter Einsatz von Entwicklerdokumentation

Vorgelegt von Kai Oey

Zusammenfassung

Dokumentation ist ein „Übel“ der Softwareentwicklung. Durch eine Effizienzsteigerung von Entwicklersoftware wird dazu beigetragen, den Erstellungsprozess von Software effizienter zu gestalten.

Entwicklerdokumentation wird für verschiedene Gegenstände der Softwareentwicklung angefertigt. Dazu zählen beispielsweise Anforderungen, Architekturen, Quellcode, Schnittstellen, Modelle und Testfälle. Es gibt einige Gründe, warum Entwicklerdokumentation erstellt werden sollte, beispielsweise Wissenstransfer, Wissenskonservierung, Kommunikation oder die Erstellung als Grundlage eines Vertrages. Entwicklerdokumentation bringt aber auch Probleme mit sich. Hier ist beispielsweise der Aktualisierungsaufwand, eine mangelnde Verständlichkeit und geringe Aktualität hervorzuheben.

Es sind seit einiger Zeit Maßnahmen bekannt, durch die die Effizienz einer Dokumentation verbessert werden kann. Unter diese Maßnahmen fallen dokumentationsbezogene (z.B. integrierte Dokumentation, hohe Lesbarkeit), organisationsbezogene (z.B. die Gestaltung der Teamgröße) und technikbezogene Konzepte zur Steigerung der Effizienz. Großen Einfluss auf die Effizienz einer Entwicklerdokumentation hat der angestrebte bzw. erstellte Umfang der schriftlichen Dokumentation. Diese kann einerseits durch eine geschickte Bestimmung der zu dokumentierenden Gegenstände, andererseits durch die Festlegung angemessener Umfänge einzelner Dokumentationen erreicht werden. Entwicklerdokumentation kann aber auch auf alternativen Wegen effizienter gestaltet werden. In der Arbeit werden dazu sieben Maßnahmen vorgestellt. Darunter fallen beispielsweise ‚Pair Programming‘, ‚direkte Kommunikation‘, ‚kontinuierliche Integration‘ und ‚öffentliche Präsentation‘.

Alle vorgestellten Maßnahmen eignen sich jedoch nicht immer alle für jedes Softwareentwicklungsprojekt. Vielmehr werden diese von verschiedenen Faktoren, die in der Softwareentwicklung auftreten können, beeinflusst und teilweise sogar determiniert. Beispielsweise kann bei hoher Dynamik von Anforderungen verstärkt ‚direkte Kommu-

nikation', ,umfassende Partizipation' und ,öffentliche Präsentation' zur Effizienzsteigerung eingesetzt werden. Diese Maßnahmen decken dann die Gründe ab, aus denen schriftliche Dokumentation erstellt worden wäre. Eine determinierende Wirkung ist beispielsweise bei hohen Sicherheitsanforderungen an Software gegeben. In diesem Fall kann nicht ohne weiteres auf schriftliche Dokumentation verzichtet werden. Verschiedene Faktoren können dabei Einflüsse auf (1) die Entwicklerdokumentation selber, auf (2) vorgestellte Maßnahmen zur Effizienzsteigerung der Dokumentation und auf (3) die Wirkung verschiedener Maßnahmen haben.

Zusammenfassend lässt sich feststellen, dass Entwicklerdokumentation effizienter gestaltet werden kann. Man muss jedoch bei einer Gestaltung die Umwelt und Situation, in der die Softwareentwicklung und das Anwendungsgebiet der Software eingebettet sind, beachten.

Kai Oey

Diplomarbeit

im Fach Management der Softwareentwicklung

Dokumentation in der Softwareentwicklung

Effizienter Einsatz von Entwicklerdokumentation

Vorgelegt in der Diplomprüfung im Studiengang Wirtschaftsinformatik der
Wirtschafts- und Sozialwissenschaftlichen Fakultät der Universität zu Köln.

Köln, im September 2002

Dokumentation ist nicht Teil der Lösung,
sondern Teil des Problems.

Tom DeMarco („Wien wartet auf Dich“, S. 130)

Inhaltsverzeichnis

Inhaltsverzeichnis	III
Abkürzungsverzeichnis	V
Abbildungsverzeichnis	VI
1 Einleitung	1
1.1 Problemstellung	2
1.2 Einordnung und Ziel der Arbeit	3
1.3 Vorgehensweise	4
1.4 Erläuterung verwendeter Begriffe	5
2 Entwicklerdokumentation	9
2.1 Gegenstände der Entwicklerdokumentation	9
2.2 Gründe für Entwicklerdokumentation	12
2.2.1 Wissenskonservierung	12
2.2.2 Wissenstransfer	13
2.2.3 Koordination	14
2.2.4 Kommunikation	15
2.2.5 Softwareerstellungs-Vertrag	15
2.2.6 Organisatorische Politik und Macht	16
2.2.7 Strukturierung der Implementierung	17
2.2.8 Durchdringung inhaltlicher Konzepte	17
2.2.9 Geforderte Qualitätsstandards	18
2.2.10 Dokumentationsvorgaben	18
2.3 Nachteile von Entwicklerdokumentation	19
2.3.1 Aktualität	19
2.3.2 Aktualisierungsaufwand	20
2.3.3 Verständlichkeit	21
2.3.4 Informationsfindung	21
2.3.5 Zusätzlicher Ressourcenbedarf	22
2.3.6 Motivationsproblem	22
3 Effizienzsteigerung des Einsatzes von Entwicklerdokumentation	24
3.1 Effizienz von Entwicklerdokumentation	24
3.2 Maßnahmen zur Effizienzsteigerung	27
3.2.1 Grundkonzepte und Qualitätsanforderungen	28
3.2.2 Quantitative Aspekte	36
3.2.3 Alternativen zu traditionellen Ansätzen der Dokumentation	42
3.3 Kritische Beleuchtung weiterer und genannter Vorschläge	53
3.3.1 Zeitpunkt von Dokumentation	53
3.3.2 Technische Hilfsmittel	54
3.3.3 Video- und Audioaufnahmen als Dokumentation	55
3.3.4 Kompletter Dokumentationsverzicht	56
3.3.5 Positive Aspekte von Redundanzen	56
3.3.6 Unnötigkeit richtiger und aktueller Dokumentation	56
3.3.7 Teamgröße und angemessene direkte Kommunikation	57
3.3.8 Öffentliche Präsentation und Prozessdokumentation	57
4 Einflussfaktoren auf Entwicklerdokumentation	58
4.1 Organisatorische Einflussfaktoren	58

4.1.1	Ablauforganisation	59
4.1.2	Aufbauorganisation	59
4.1.3	Umfang eines Projekts	60
4.1.4	Zeitraum der Erstellung eines Projektes	61
4.1.5	Fluktuation	62
4.1.6	Arbeitsteilung	62
4.1.7	Planung und Kontrolle	63
4.1.8	Entscheidungsdelegation	64
4.1.9	Vorauskoordination	65
4.1.10	Formalisierung	65
4.2	Softwarespezifische Einflussfaktoren	66
4.2.1	Art der Auslösung	66
4.2.2	Unklarheit der Entwicklungsaufgabe	67
4.2.3	Dynamik der Anforderungen	68
4.2.4	Neuartigkeit der Technologie	69
4.2.5	Intensität der Kunden-/Marktkommunikation	69
4.2.6	Sicherheitsanforderungen der Software	69
4.2.7	Lebensdauer von Entwicklerdokumentation und Software	70
4.2.8	Vorgehensmodelle	71
4.3	Weitere Einflussfaktoren	72
4.3.1	Verhalten der Mitarbeiter	72
4.3.2	Ausbildung und Wissen der Mitarbeiter	72
4.3.3	Technische Unterstützung der Entwicklung	73
4.3.4	Wirtschaftlicher Druck auf Software-Firmen	74
5	Schlussbetrachtung	75
6	Literaturverzeichnis	76
7	Anhang A: Literatur zu Formatierungsvorschlägen von Code und Kommentaren	92