Dokumentation für die Software-Wartung

Stefan Opferkuch

Abteilung Software Engineering, Institut für Softwaretechnologie
Universität Stuttgart

www.iste.uni-stuttgart.de/se

1 Einführung

Es gibt verschiedene Auffassungen darüber, was Software-Wartung ist, wann sie beginnt und welche Teilaufgaben dazugehören. Unbestritten ist lediglich ihre große Bedeutung als längste Aktivität im Softwarelebenszyklus. So gilt nach Lehmans erstem Gesetz der Software-Evolution, dass ein Softwaresystem kontinuierlich an eine sich verändernde Umgebung angepasst werden muss (Lehman und Belady, 1985). Nur so kann verhindert werden, dass das System veraltet.

Um den Begriff Software-Wartung zu klären, werden zunächst einige Definitionen vorgestellt und diskutiert. Nach dem IEEE Standard 1219 (IEEE 1219-1998) gehören zur Software-Wartung alle Änderungen eines Softwareprodukts, die nach dessen Auslieferung durchgeführt werden, um Fehler zu verbessern, die Leistungsfähigkeit zu erhöhen oder das Produkt einer veränderten Umgebung anzupassen. Der ISO-Standard 12207 (ISO/IEC 12207-1995) beschreibt Software-Wartung als Änderungen der Software, die auf Grund eines Problems oder Verbesserungsbedarfs notwendig werden. Dabei werden kleine Teile der Software verändert, jedoch bleibt der Großteil der Funktionalität erhalten.

Nach diesen Definitionen wird die Software-Wartung ausschließlich nach der Implementierung durchgeführt. Diese Sichtweise wird jedoch zunehmend in Frage gestellt. So formuliert Pigoski (1997), dass Wartung zeitgleich mit der Entscheidung beginnt, ein Softwaresystem zu entwickeln.

In diesem Artikel wird folgende Definition der Software-Wartung verwendet: Software-Wartung ist jede Arbeit an einem bestehenden Software-System, die nicht von Beginn der Entwicklung an geplant war oder hätte geplant werden können und die unmittelbare Auswirkungen auf den Benutzer der Software hat (Ludewig und Opferkuch, 2004).

2 Warum wird Software-Wartung notwendig?

Software-Wartung kann aus verschiedenen Gründen notwendig werden, z. B. um neue Funktionalität in die Software einzubauen, Fehler zu beheben, das Systemdesign zu verbessern oder den Ressourcenverbrauch zu senken (Feathers, 2004).

Swanson (1976) beschreibt drei Kategorien, in die sich Änderungen im Rahmen der Software-Wartung klassifizieren lassen: korrektive, adaptive und perfektive Wartung. Diese drei Kategorien wurden durch die IEEE (1990) um präventive Wartung erweitert.

3 Warum ist Software-Wartung so schwierig?

Software-Wartung unterscheidet sich von der Wartung technischer Produkte.

Während die Wartung technischer Produkte auf die möglichst lange und perfekte Erhaltung der Produkte abzielt, dient die Software-Wartung dazu, die Software veränderten Rahmenbedingungen anzupassen.

Das Ziel der Software-Wartung ist also nicht das Erhalten eines originalen Zustands, sondern im Gegenteil das Herstellen eines neuen Zustands, damit die Software einsetzbar und nützlich bleibt. Dafür sind häufig aufwändige und potentiell fehlerträchtige Änderungen der Struktur notwendig.

Die Software-Wartung ist nicht die Fortsetzung der Entwicklung mit anderen Mitteln!

Es gibt einen entscheidenden Unterschied: bei der Software-Entwicklung ist typischerweise nur eine begrenzte Anzahl an Einschränkungen für das neue System zu beachten, wogegen die Wartung im Kontext eines bestehenden und sich im Einsatz befindlichen Systems durchgeführt werden muss. Aus diesem Grund müssen alle Änderungen bezüglich ihrer Auswirkungen auf das gesamte System hin geprüft werden.

Zudem haben die Wartungsingenieure des Systems dieses in der Regel nicht selbst entwickelt und erreichen kaum einen vollständigen Überblick darüber.

Software-Wartung muss schon während der Entwicklung berücksichtigt werden.

In der Entwicklung müssen sowohl organisatorische, wie auch technische Maßnahmen für die Wartung ergriffen werden. Zu den organisatorischen Maßnahmen gehört, dass schon zu Beginn der Entwicklung ein Configuration Management eingesetzt werden muss. Ein nachträgliches Installieren am Ende der Entwicklung ist nur unter großem Aufwand und hohen Kosten möglich. Ähnliches gilt für das Change Management. Bereits zu Beginn der Entwicklung muss ein tragfähiges Vorgehen umgesetzt werden, wie mit Änderungen der Software umgegangen wird.

Zu den technischen Maßnahmen gehört die Definition und Umsetzung von Codierrichtlinien sowie die Verwendung von Entwurfsmustern und Frameworks, um eine klar strukturierte und nachvollziehbare Software zu erstellen. Darüber hinaus kann die

Wartung durch die Verwendung von COTS-Produkten und Bibliotheken vereinfacht werden.

Schließlich sind verschiedene Dokumente für die effiziente Software-Wartung von zentraler Bedeutung. Es wird eine Gesamtbeschreibung des Systems und eine Erläuterung der Abhängigkeiten zwischen den Teilen des Systems benötigt. Der Aufwand für die nachträgliche Erstellung dieser Dokumente ist deutlich höher als die zeitnahe Erfassung während der Entwicklung.

4 Dokumentation für die Wartung

Eine sinnvolle und aktuelle Dokumentation unterstützt den Wartungsingenieur bei der Durchführung von Änderungen. Er gewinnt schnell einen Überblick, findet die Funktionalität im Quellcode und kann Auswirkungen von Änderungen abschätzen.

Aktueller Stand der Dokumentation

Kajko-Mattsson (2001) hat den Stand der Dokumentation in der Wartung untersucht. Dazu wurden 19 Abteilungen in 18 schwedischen Unternehmen befragt. Ergebnis waren unter anderem folgende Punkte:

- 10 Abteilungen beginnen die Software-Wartung mit vollständigen und konsistenten Entwicklungsdokumenten. Nur 3 aktualisieren alle Dokumente während der Wartung.
- 8 Abteilungen legen Codierrichtlinien fest; 4 geben Richtlinien vor, wie Systemdokumentation verfasst werden soll.
- 5 Abteilungen haben Checklisten, welche Dokumente bei der Wartung vorhanden sein bzw. erstellt werden müssen; in nur 4 Abteilungen wird festgelegt, zu welchem Zeitpunkt im Prozess welche Art von Dokumenten überarbeitet oder überprüft werden sollen.
- 5 Abteilungen führen eine formale Abnahme der Systemdokumentation nach jeder Wartungsaktivität durch.

Aus diesen Zahlen leitet Kajko-Mattsson ab, dass die Dokumentation in der Wartung ein noch immer stark vernachlässigter Aspekt ist und die Software-Wartung dadurch erschwert wird.

Art und Umfang der Dokumentation

Da Software-Wartung häufig über einen langen Zeitraum und unter der Beteiligung vieler Personen durchgeführt wird, besteht ein hoher Informationsbedarf darüber, warum einzelne Änderungen durchgeführt wurden. Es ist sehr wichtig, dass die erfassten Informationen aktuell gehalten werden, da eine veraltete und evtl. inkonsistente Dokumentation das Verständnis der Software erschwert. Eine hochwertige und aktuelle Dokumentation ist Voraussetzung für eine effiziente Software-Wartung.

Als Grundlage für die Wartung werden aus der Entwicklung eine *Systembeschreibung* und eine *codebezogene Dokumentation* benötigt. Die Systembeschreibung erfolgt während der Entwicklung in einer Spezifikation und einem Architekturentwurf, die codebezogene Dokumentation in einem Modulentwurf oder als interne Dokumentation im Code selbst. Vor allem bei der Systembeschreibung ist am Ende der Entwicklung darauf zu achten, dass diese den aktuellen Zustand des Systems beschreibt.

Die Entwicklungsdokumentation bildet die Basis der Wartungsdokumentation, sie reicht jedoch nicht aus. Zusätzlich müssen für die Software-Wartung die *Abhängigkeiten* sowohl innerhalb des Quellcodes als auch zwischen den Anforderungen und deren Umsetzung erfasst werden. So kann der Wartungsingenieur schnell Zusammenhänge zwischen den Anforderungen und dem Code nachvollziehen.

5 Aktuelle Arbeiten

Eine gute, umfassende Dokumentation für die Wartung ist teuer und somit als universelle Lösung für jede Art von Wartung ungeeignet. Aus diesem Grund arbeiten wir an der richtigen Art und dem richtigen Umfang der Dokumentation für die Software-Wartung. Dabei wird der Frage nachgegangen, welche Dokumentation welcher Softwareteile unter Kosten-Nutzen-Aspekten notwendig und sinnvoll ist.

Literatur

FEATHERS, M.C.: *Working Effektively with Legacy Code*. Prentice Hall PTR, Upper Saddle River, NJ 07458, 2004.

IEEE – THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.: *IEEE Std. 610.12-1990: Standard Glossary of Software Engineering Terminology*. New York, NY.

IEEE – THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.: *IEEE Std. 1219-1998: IEEE Standard for Software Maintenance*. New York, NY.

ISO – INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *(ISO/IEC 12207) Standard for Information Technology – Software Lifecycle Processes*. Genf, Schweiz, 1995.

KAJKO-MATTSSON, M.: *The State of Documentation Practice within Corrective Maintenance*. Erschienen in: Proceedings of the ICSM, Florenz, Italien, 2001.

LEHMAN, M.M. und L.A. BELADY: *Program evolution: processes of software change*. Academic Press Professional, Inc., San Diego, CA, USA, 1985.

LUDEWIG, J. UND S. OPFERKUCH: *Software-Wartung – eine Taxonomie*. Gesellschaft für Informatik e.V., Softwaretechnik-Trends, Band 24 Heft 2, Mai 2004.

PIGOSKI, T.M.: *Practical Software Maintenance: Best Practices for Managing your Software Investment*. New York, N. Y. [u.a.], Wiley, 1997.

SWANSON, E.B.: *The Dimensions of Maintenance*. Erschienen in: 2nd International Conference on Software Engineering, San Francisco, CA, USA, IEEE Computer Soc. Pr., 1976.