

13 Softwarewartung und Prozessmodelle in Theorie und Praxis

Urs Kuhlmann

Andreas Winter

Universität Koblenz-Landau, Institut für Softwaretechnik, D-56016 Koblenz

(kuhlurs|winter)@uni-koblenz.de

13.1 Motivation und Wartungsbegriff

Sowohl der Begriff der *Wartung* als auch der des *Reengineering*s sind spätestens seit den 1980er Jahren im Bereich der Softwaretechnik in der Literatur und in der Praxis etabliert. Allerdings gibt es bisher keine einheitlichen und allgemein gebräuchlichen Begriffsdefinitionen.

Wartung wird in den meisten Definitionen ausdrücklich als Aktivität „nach Auslieferung des Softwareprodukts“ verstanden. Die meisten Autoren betonen aber auch, dass die Planung von Wartungsaktivitäten mit der Planung der Softwareentwicklung beginnen soll. Wartungsaktivitäten sind daher nicht losgelöst von Software-Entwicklungsprozessmodellen zu betrachten, sondern sollten als zentrale Bestandteile der Software-Entwicklung verstanden werden.

Diese Arbeit beschreibt die Einbettung der Wartung in Prozessmodelle der Theorie und fasst sechs Fallstudien zusammen, die die Integration der Wartungsaktivitäten in die betriebliche Praxis zeigen. Eine Zusammenfassung mit einem Ausblick über mögliche Lösungsansätze zur verbesserten Etablierung der Wartung in der Softwareentwicklung beendet diese Kurzfassung.

13.2 Prozessmodelle

In den Prozessmodellen der Softwareentwicklung sind Wartungsaktivitäten sehr unterschiedlich integriert. Dieser Abschnitt liefert eine Übersicht über die Einbettung von Wartungsaktivitäten in verschiedenen Prozessmodellen.

- Im *Wasserfallmodell* [1], als Beispiel für lineare Modelle, sind Wartungsaktivitäten in einer separaten, nämlich der letzte Phase zusammengefasst. Diese Phase wird als Betrieb und Wartung an die eigentliche Entwicklung angehängt.
- Der *IEEE Standard for Software Maintenance* [3] liefert eine Beschreibung für die Vorgehensweise bei Wartungsprojekten. Er beschreibt detailliert linear angeordnete Wartungsaktivitäten. Problematisch ist hier die Tatsache, dass nur ein einziger, sehr umfangreicher und grober Rahmen geliefert wird. Kleinere, speziellere Wartungsaufgaben werden aufgrund der Allgemeinheit dieses Raster nicht ausreichend unterstützt.
- In *Extreme Programming* [2], als Beispiel für die agile Software-Entwicklung, wird nur erwähnt, dass Wartung als Normalzustand eines XP Projektes angesehen werden kann. XP ist als generelles Prozessmodell für Wartung im Allgemeinen nur beschränkt einsetzbar. Auch beschränken Projektgröße und -ziele den Einsatz von XP im Reengineering [6].

- Im *(Rational) Unified Process* [4], einem inkrementellen Modell, sind wartungsrelevante Aktivitäten zwar in vielen Workflows enthalten. Jedoch wird Wartung nicht explizit unterstützt und genutzte Aktivitäten sind eher auf Neuentwicklung ausgerichtet. Wichtige analytische Wartungsaktivitäten wie z. B. das Programmverstehen und die Impact-Analyse sind im RUP nicht enthalten.

Wartung wird nur im Wasserfallmodell explizit erwähnt und ist in anderen Modellen i.d.R. implizit enthalten. Die meisten Aktivitäten werden nicht unter Wartungsaspekten betrachtet, reine Wartungsaktivitäten fehlen. Moderne Prozessmodelle decken Wartung gar nicht oder nur ungenügend ab [5].

13.3 Fallstudien

Eine Befragung von sechs Firmen zum Thema Softwarewartung [5] ergab, dass auch in der Praxis kein einheitlicher Wartungsbegriff verwendet gibt. Die Begriffsverwendungen sind im Vergleich zur Literatur sogar noch verschwommener, was deutlich ausgedrückt wird durch Aussagen wie „Wir warten keine Software, wir entwickeln sie weiter“ und „Wartung ist der Weg vom Ist zum Soll“.

Gängige Praxis zur Definition von Wartung ist die Eingrenzung über den Aufwand, der zur „Änderung“ aufgewandt wird. Wartung umfasst z. B. alle Änderungen mit einem Aufwand von weniger als 15 oder 30 Personentagen. Jedoch wird dadurch Wartung hauptsächlich auf korrektive Wartung beschränkt. Wartungskategorien wie adaptive, perfektive und preventive Wartung werden (wenn überhaupt) als von (korrektiver) Wartung unabhängig betrachtet. Wartung umfasst jedoch mehr als nur das Beheben von Fehlern. So werden spezifische Wartungsaktivitäten, besonders analytische Aktivitäten, auch bei beliebigen Anpassungen bestehender Programmen an geänderte Umgebungen benötigt.

Wartung ist nur selten in die Entwicklungsprozesse der Firmen integriert. Meist werden Wartungsaktivitäten separat betrachtet und in getrennten Wartungsprojekten durchgeführt. Die ausgeführten Aktivitäten sind dabei teilweise identisch, aber zumindest sehr ähnlich zu denen der Entwicklung. So wird die Aktivität „Implementierung“ in der Entwicklung als auch in der Wartung durchgeführt. Es werden aber auch Aktivitäten benötigt, die in reinen Entwicklungsprojekten nicht vorkommen.

Wesentliche Faktoren, die Wartungsprozesse beeinflussen, sind die Firmenkultur und das zu wartende System selbst. Auch die Umsetzung von Programmierrichtlinien bei der Entwicklung und in früheren Wartungsprojekten sind entscheidend. Weiterhin wurden die Fachkenntnisse

der Beteiligten und eine angemessene Werkzeugunterstützung hervorgehoben.

Auch wenn sich die Befragten gute Noten für die Qualität ihrer Wartungstätigkeiten geben, erhoffen sie sich Qualitätsverbesserungen durch Veränderung ihrer Prozesse. Verbesserungen werden durch ausgereifere und besser integrierte Werkzeugen erwartet. Ursachen für die bisher geringe Werkzeugnutzung liegen in den Besonderheiten zu wartender Software und dem Anpassungsaufwand der vorhandenen Werkzeuge. Aufgrund eher schrumpfender Wartungsbudgets wird allerdings nicht erwartet, dass die erhofften Änderungen in naher Zukunft umgesetzt werden.

Die wesentlichen Ergebnisse der Fallstudien sind, dass der Wartungsbegriff auch in der Praxis nicht klar definiert ist, Wartungsaktivitäten nur bedingt in betriebliche Softwareentwicklungsprozesse integriert sind und dass ein wesentlicher Änderungswunsch die Werkzeugunterstützung für den Bereich der Wartung betrifft [5].

13.4 Zusammenfassung und Ausblick

Aus der Untersuchung des Wartungsbegriffs in Theorie und Praxis lassen sich einige Problembereiche für die unzureichende Wahrnehmung von Fragestellungen der Wartung ableiten. Im Folgenden werden diese Problembereiche sowie einige Lösungsansätze zur besseren Etablierung der Wartung in der Software-Entwicklung skizziert [5].

Terminologie

Die Begriffsinhalte der Wartung sind weder in der Literatur noch in der Praxis eindeutig definiert und werden daher vielseitig verwendet. Auch eine Abgrenzung von Neuentwicklung und Wartung ist schwierig und kaum möglich. Es gibt keinen klaren Übergang von Entwicklung zu Wartung, denn Wartungsaktivitäten müssen bereits bei der Entwicklung berücksichtigt und eingeplant werden. In gemeinsamen Projekten ist es daher notwendig eine *gemeinsame Definition* als Arbeitsgrundlage zu finden.

Statistische Daten

Statistische Daten über Wartung liegen nur aus den 1970er und 1980er Jahren vor. Aktuelle Angaben über die Kosten der Wartung und die Aufteilung auf verschiedene Wartungskategorien sind nicht verfügbar. In den meisten Firmen erfolgt keine Abgrenzung zwischen Entwicklungs- und Wartungsbudgets. Zur Ermittlung der Bedeutung von Wartung sind *empirische Untersuchungen* zur Sammlung von aktuellen statistischen Daten notwendig.

Bewusstsein für Wartung

Wartung wird in der Literatur zwar oft als ein wichtiges Gebiet bezeichnet, aber der Schwerpunkt der Aufmerksamkeit liegt auf der Neuentwicklung. Auch in der Praxis ist die Wartung nur das ungeliebte Stiefkind, dem keine Priorität gegeben wird. Es wird, im Gegensatz zu Empfehlungen aus der Literatur, auch oft als Trainingsbereich für

neue Mitarbeiter eingesetzt, die bei entsprechender Leistung „vom Wartungspersonal zu Entwicklern aufsteigen können“. Daher ist es notwendig, den Entscheidungsträgern in Firmen die *Bedeutung und Folgen* von (mangelhafter) Wartung zu verdeutlichen und bei Mitarbeitern für Wartungsprojekte zu werben.

Einbettung der Aktivitäten in Prozessmodelle

Wartungsaktivitäten werden von existierenden Prozessmodellen nur ungenügend abgedeckt. Auch viele Firmen haben nur einen beschriebenen Entwicklungsprozess und keinen klar definierten und praktizierten Wartungsprozess. In diesem Bereich ist es nötig, wartungsspezifische Aktivitäten zu identifizieren und zu beschreiben. Zur Berücksichtigung in Software-Entwicklungs- und -Wartungsprozessen müssen diese in bestehende Prozessmodelle eingebettet oder durch ein *Referenzprozessmodell* für Softwarewartung zur Verfügung gestellt werden. Dieses kann z. B. durch Zusammenfassung der für die Wartung relevanten Aktivitäten in einen Wartungsworkflow des RUP erfolgen.

Werkzeugunterstützung

Eine umfangreiche Werkzeugunterstützung wird in der Literatur als kritischer Faktor für die Erhöhung von Qualität und Effizienz genannt. Obwohl eine stärkere Unterstützung in der Praxis gewünscht wird, ist sie nur selten anzutreffen. Ein mögliches Referenzprozessmodell für die Softwarewartung sollte um passende *Werkzeugunterstützung* erweitert werden, damit die Wartungsaktivitäten effizient und mit hoher Qualität durchgeführt werden können.

13.5 Fazit

Zusammenfassend bleibt festzustellen, dass sich die Software-Wartung als *zentrale Disziplin der Software-Entwicklung* noch etablieren muss. Die klare Definition des Wartungsbegriffes und die Abgrenzung zur Entwicklung ist hierbei aber nicht das Wesentliche. Vielmehr ist die Einbettung von Wartungsaktivitäten in Prozessmodelle zur Software- (Weiter-) Entwicklung und die konsequente werkzeuggestützte Anwendung dieser Aktivitäten der wesentliche Erfolgsfaktor für die zukünftige Softwareerstellung.

Literaturverzeichnis

- [1] W.W. Agresti. The Conventional Software Life-cycle Model: Its Evolution and Assumptions. *New Paradigms for Software Development*, pp 2–5, 1986.
- [2] K. Beck. *Extreme programming explained: embrace change*. Addison-Wesley, Upper Saddle River, 2000.
- [3] IEEE Standard for Software Maintenance, IEEE Std 1219-1998. The Institute of Electrical and Electronics Engineers, 1998.
- [4] P. Kruchten. *The Rational Unified Process - An Introduction*. Addison-Wesley, Upper Saddle River, 2nd edition, 2000.

[5] U. Kuhlmann. Maintenance Activities in Software Process Models: Theory and Case Study Practice. Universität Koblenz, 2004.

[6] C. Poole and J.W. Huisman. Using Extreme Programming in a Maintenance Environment. *IEEE Software*, 18(6):42–50, 2001.