

Kennzahlenbasierte Sanierung von Softwaresystemen

Eberhard Kuhn und Jochen Winzen, andrena objects ag

Gegenstand dieses Artikels ist die kostengünstige Sanierung objektorientierter Softwaresysteme zur nachhaltigen Sicherung des ROI. Um dies im laufenden Betrieb zu ermöglichen, hat andrena ein eigenes Werkzeug entwickelt (ISIS), das die Steuerung und Transparenz während der Sanierungsphase gewährleistet.

Vor 36 Jahren wurde der Begriff Softwarekrise geprägt, der die Probleme mit der termingerechten Erstellung und der allzu häufig mangelnden Qualität von Programmen adressiert. Fehlende Qualität führt im Lebenszyklus eines Programms zu schneller Degeneration. Die Wartbarkeit und Erweiterbarkeit stößt zunehmend an Grenzen und macht eine Neuimplementierung erforderlich.

Vergleicht man die Komplexität von Programmen im Jahr 1972 mit der heutigen, liegen Welten dazwischen. Die Probleme sind geblieben, offenbaren sich aber auf anderem Niveau. Eine der erfolgreichsten Innovationen zur Beherrschung der Komplexität ist Objektorientierte Programmierung. Das Programm wird von der Makroebene absteigend in Module (Klassen) strukturiert, die eine schlanke Schnittstelle besitzen und die Daten sowie die darauf arbeitende Logik kapseln.

Die Softwarekrise im Jahr 2008

Die Softwarekrise ist dennoch bis zum heutigen Tag nicht beendet. Zum einen wird das Potential Objektorientierter Programmierung nur ansatzweise ausgeschöpft, zum anderen bestehen nach wie vor Probleme, die letztendlich mit der mangelnden Umsetzung des Software Engineering (Ausbildung, Erfahrung, Anwendung des Standes der Technik) zu tun haben. Dies wird daran deutlich, dass viele Softwaresysteme bereits nach kurzer Lebensdauer – schlimmstenfalls nach Primärimplementierung - Sanierungsfälle sind.

Häufig werden Programme innerhalb einer kurzen Zeitspanne in Grund und Boden versenkt: Die mangelhafte Bauweise (Architektur, Design, Codierung) führt selbst bei kleinen Änderungen zu erheblichen Nebenwirkungen und Bugs, der Releasetermin naht, in Nacht- und Nebelaktionen wird quick & dirty ein für den Anwender halbwegs akzeptables System notdürftig zusammengezimmert. Die Gegenwart wird von diesem Anwendungsniveau geprägt: Pfusch!

Worin äußert sich der Sanierungsfall

Für Anwender oder das Management am leichtesten erkennbar sind Probleme der äußeren Qualität (Korrektheit, Bedienbarkeit, Performanz, Skalierbarkeit). Das Programm bringt nicht die vom Anwender erwartete Funktionalität.

Probleme der inneren Qualität (Bauweise, also die Verständlichkeit, Wartungs-, Erweiterungs- und Änderungsfreundlichkeit) äußern sich in zunehmenden Kosten bei Änderungen und Erweiterungen. Sie schlagen auf die äußere Qualität durch (langwierige Erweiterungen, viele Bugs bei Änderungen). Bestünden keine Probleme der inneren Qualität, könnten Mängel der äußeren Qualität einfach behoben werden.

Warum Software zum Sanierungsfall wird

- Struktur und Implementierung entsprechen nicht dem Stand der Technik
- Funktionsgetrieben: Fokus auf äußerer Qualität – Vernachlässigung der inneren Qualität
- Die Beherrschung der Syntax einer objektorientierten Sprache bedeutet nicht die Beherrschung der Konzepte objektorientierter Programmierung
- Automatische Migration eines Altsystems auf neue Sprache (z. B. Migration Visual Basic nach C#)
- Entwickler folgen nicht den schnellen Innovationszyklen, Neuerungen werden in der Software nicht mitgezogen (z. B. automatisierte Tests)
- Unangemessener Einsatz von Technologien (Overdesign)
- Zu hohe Fluktuation im Projektteam (technisches Management) : Baustellen werden nicht geschlossen, Erosionskeime und Verwerfungen entstehen

Defizite bei der Verbesserung von Problemsoftware

An erster Stelle stehen Ignoranz und Verdrängung: „die Software läuft doch“. Selbst falls Probleme erkannt werden, wird die Kosteneinsparung durch eine Sanierung unterschätzt. Dies liegt an folgenden Aspekten:

- Die Verbesserung der inneren Qualität ist für das Management bzw. den Kunden nicht sichtbar
- Es gibt daher keine Wertschätzung für solche Maßnahmen
- Für Softwareentwickler ist die Verbesserung der inneren Struktur eine undankbare Aufgabe: Legacy Code ist schwer verständlich, die Restrukturierung ist riskant.

Die Sanierungsmaßnahmen sind eher willkürlich, an Bugs orientiert. Werkzeuge zur Qualitätsanalyse werden nur sehr vereinzelt eingesetzt. Summa summarum liegt daher häufig keine Sanierung, sondern ein Herumdoktern an Symptomen vor (wieder Pfuscher).

Ziele der Sanierung

Im Zentrum steht die Sicherung des ROI: die Reduzierung der Kosten für Änderungen und Erweiterungen des Programms, die Vermeidung einer kostenträchtigen Neuimplementierung.

Kurzfristiges Ziel ist die Stabilisierung der Funktionalität (Bugs) und die Implementierung der drängendsten Neuanforderungen.

Mittelfristig steht dann die Abarbeitung des Problemstaus (Hotspots, neue Anforderungen) und der Transfer von Know How in das Entwicklungsteam im Fokus.

Langfristiges Ziel ist die Stabilisierung bezüglich Kosten und Qualität. Zu guter Letzt ist die Schaffung von Entscheidungsspielraum zu nennen.

ISIS – Steuerung und Transparenz

ISIS ist ein von der andrena objects ag konzipiertes Navigationssystem zur Steuerung der Prozess- und Softwarequalität und zur Schaffung von Transparenz. Ziel ist die am Kosten/Nutzen-Optimum orientierte Softwareentwicklung und Softwaresanierung, die Sicherung des ROI.

Hierfür werden wenige Indikatormetriken in regelmäßigen Abständen automatisiert erfasst. Sie stehen für die Qualität der Architektur, des Designs und der Codierung. Die Rohmetriken werden auf eine Qualitätsskala von 0 (sehr schlecht) bis 100 (sehr gut) projiziert und auf 2 Kenngrößen zur Prozess- und Softwarequalität (Qualitätsindizes) verdichtet. Der Zeitaufwand für die Messung beträgt etwa ein Promille der Arbeitszeit.

Der entscheidende Vorteil liegt in dem Gewinn an Transparenz: die Qualitätsdaten dienen dem Entwicklungsteam zur Überwachung des Projektfortschritts und zur Identifizierung von Defiziten. Das Management hat jederzeit objektivierte Kenngrößen für Projektfortschritt, Prozessqualität, äußere Qualität und innere Qualität. Qualitätsverbesserung aber auch Verschlechterungen werden sichtbar. Das System ist in allen Java- und C#-Projekten einsetzbar.

Mit ISIS können Softwaresysteme sehr effizient auf ihre Qualität hin analysiert werden, in der Regel benötigt man nur wenige Stunden, um dezidierte Aussagen und Handlungsempfehlungen geben zu können. Liegt der Softwarequalitätsindex (SQI) kleiner 30, liegt erfahrungsgemäß ein Sanierungsfall vor.

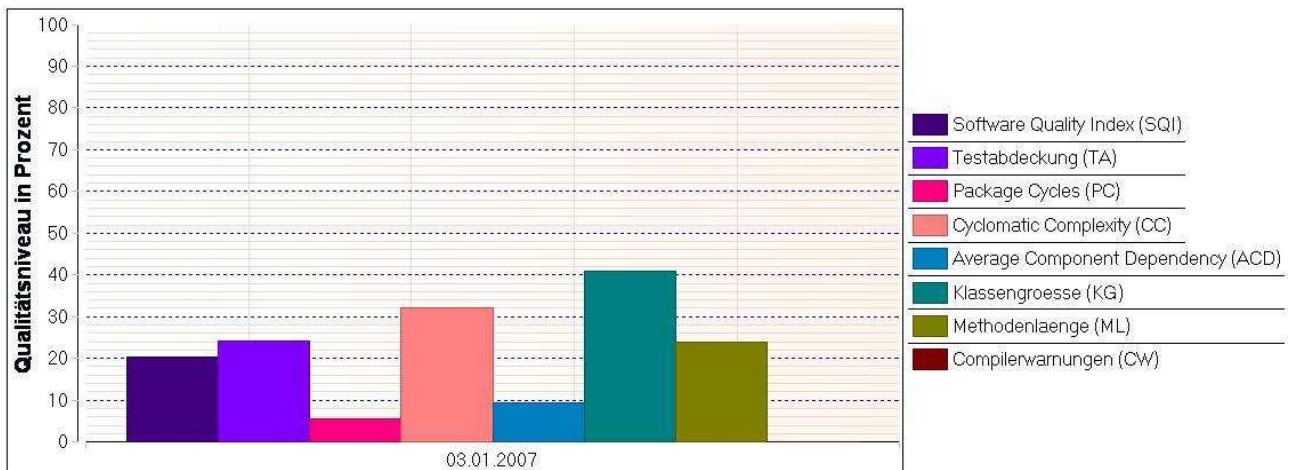


Abbildung 1: Messpunkt mit Indikatormetriken für den Softwarequalitätsindex (SQI) zu Beginn der Sanierungsphase

Vorgehensweise bei der Sanierung im laufenden Betrieb

Zu Beginn wird eine initiale Messung und Analyse der Softwarequalität (siehe Abbildung 1) sowie eine Analyse von Prozessdefiziten durchgeführt. Die Ergebnisse dienen der Erarbeitung eines Sanierungskonzeptes:

- Erarbeitung von Maßnahmen zur Etablierung des Standes der Technik
- Definition von Zielen auf Basis einzelner Indikatoren
- Festlegung von Maßnahmen zum Wissenstransfer

Die Sanierung selbst kann im laufenden Betrieb erfolgen, es ist kein Entwicklungsstopp erforderlich. Um dies zu ermöglichen ist die Etablierung des Standes der Technik unabdingbar:

- Saubere objektorientierte Programmierung
- Automatisierte Tests
- Einsatz von Werkzeugen zur Qualitätssicherung.

Experten arbeiten als Softwareentwickler direkt im Entwicklungsteam des Kunden mit. Dadurch wird eine nachhaltige Qualitätsverbesserung gewährleistet.

Die Steuerung der Sanierung erfolgt über Zielvorgaben bezüglich Qualitätsindex und einzelner Metriken (z. B. Testabdeckung) und deren regelmäßiger Überwachung. Die Qualitätsdaten werden für das Management publiziert, so dass eine weitgehende Transparenz über den Projektfortschritt gegeben ist.

Die Erstellung von automatisierten Tests läuft strukturellen Änderungen und funktionalen Erweiterungen voraus. Die zentrale Programmiermethode bei der Sanierung im laufenden Betrieb ist wie folgt:

- Implementierung automatisierter Tests für eine Programmkomponente

- Strukturelle Verbesserung der Komponente unter Wahrung der alten Funktionalität (Refactoring)
- Automatisierter Test für eine neue Teilfunktionalität
- Implementierung der neuen Teilfunktionalität

Erfahrungen

Die beschriebene Vorgehensweise bei der Sanierung im laufenden Betrieb wurde seit Anfang 2007 in mehreren Projekten erfolgreich eingesetzt. Die Beschränkung auf wenige Indikatormetriken zur Steuerung des Prozesses hat sich bewährt. Es besteht allerdings eine Gefahr zur Vernachlässigung nicht messbarer Größen (z. B. Qualität und Umfang der Dokumentation). Das Sanierungskonzept muss daher alle wesentlichen Zielgrößen umfassen und darf nicht auf einige Indikatoren eingeschränkt sein.

Die Einführung von ISIS initiiert im Entwicklungsteam eine intensive Diskussion über Softwarequalität, Entwicklungsprozesse und schafft nachhaltiges Qualitätsbewusstsein. Die Frustration des Entwicklungsteams wandelt sich durch die Transparenz der Sanierungsfortschritte in Motivation, Auftraggeber und das Management gewinnen durch die Transparenz Vertrauen.

Eine Änderung des bisherigen Entwicklungsprozesses ist in der Regel unvermeidlich. Ein Umdenken bei allen Beteiligten ist erforderlich. Dieser Lernprozess erfordert unabdingbar eine lernorientierte Fehlerkultur.

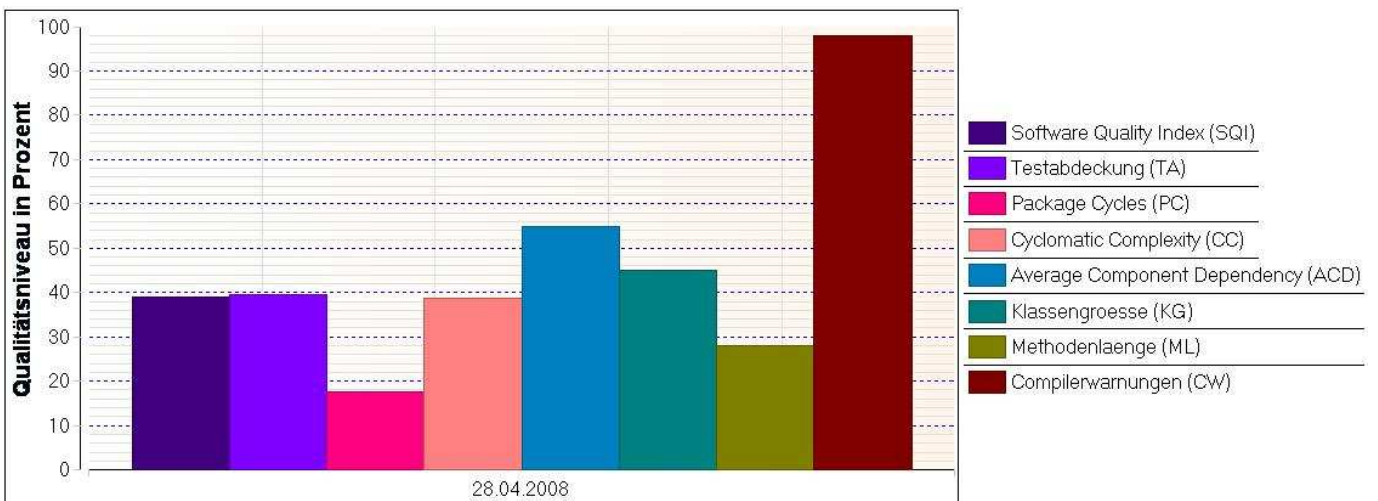


Abbildung 2: Messpunkt mit Indikatormetriken für den Softwarequalitätsindex (SQI) nach 16 Monaten Sanierung im laufenden Betrieb

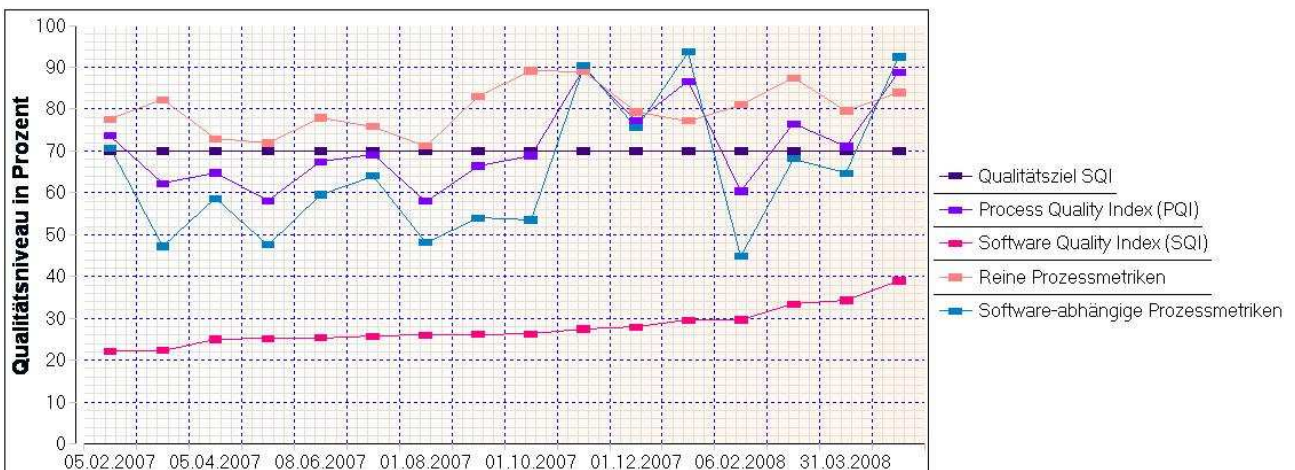


Abbildung 3: Verlaufskurven der Qualitätswerte des Sanierungsprojekts. Die rote Linie kennzeichnet die Entwicklung der Softwarequalität (SQI), die oberen Linien stellen empfindliche Sensoren für Aspekte der Prozessqualität dar.

Kurzbiografie

Dr. Eberhard Kuhn ist seit zehn Jahren bei der andrena objects ag als Softwareentwickler und Projektleiter tätig und im Rahmen der EN ISO 9001 Qualitätsbeauftragter des Unternehmens. Darüber hinaus verantwortet die Konzipierung und Entwicklung von ISIS.

Jochen Winzen ist seit 3 Jahren bei der andrena objects ag als Softwareentwickler und Projektleiter tätig sowie Qualitätsbeauftragter für Schulungen, Coaching und Reviews. Er war als Softwareentwickler an der Einbindung von Werkzeugen in ISIS beteiligt.

Kontakt Daten

Dr. Eberhard Kuhn
andrena objects ag
Albert-Nestler-Strasse 11
76131 Karlsruhe
Tel: 0721/6105-125
e-mail: eberhard.kuhn@andrena.de

Jochen Winzen
andrena objects ag
Albert-Nestler-Strasse 11
76131 Karlsruhe
Tel: 0721/6105-125
e-mail: jochen.winzen@andrena.de