20 Jahre WSR(E) – Sind wir vorangekommen oder kochen wir immer wieder die gleiche Suppe?

Podiumsdiskussion

Jens Borchers
Borchers Beratung f. Informationsmanagement, Hamburg
Email: jens@borchers-bfi.de / jensborchers@acm.org

Abstract: *Nach nunmehr 20 Jahren WSR(E) stellt sich die Frage, was wir in dieser langen Zeit an wirklichen Innovationen im Feld des Software-Reengineering, der Software-Evolution und Design-Ansätzen zur Minimierung der Wartungsaufwendungen von großen, komplexen Softwaresystemen erreicht haben?*
Machen wir noch Fortschritte oder wenden wir die lang bekannten Methoden und Techniken immer nur auf die gerade aktuellen Software-Implementierungsformen und Architekturparadigmen an?
Eine kritische Reflexion erscheint notwendig. Diese soll die Podiumsdiskussion auf dem WSRE nach 20 Jahren angehen.

1 Ziel der Diskussion

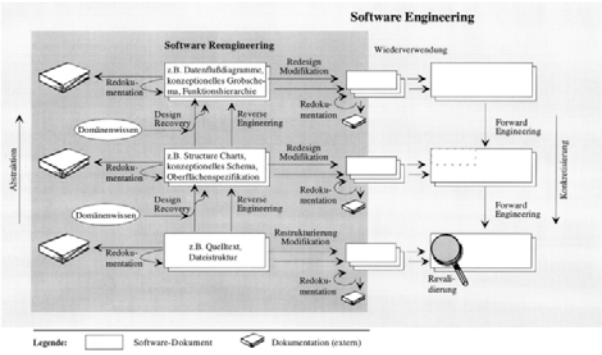
Die Disziplin „Software Engineering“ hat im vergangenen Jahr das 50-jährige Jubiläum der berühmten NATO-Konferenz in Garmisch [1] gefeiert, das wohl meistzitierte Papier zum Thema „Reverse Engineering“ [2] (mit der deutschen Adaption [3]) feiert nächstes Jahr seinen 30. Geburtstag. Auch der Workshop in Bad Honnef, der in diesem Jahr 20 Jahre alt wird, ist bereits eine Institution.
Aber was ist in diesen 20 Jahren an Neuerungen passiert? Welche bahnbrechenden Errungenschaften sind gelungen, wo kochen wir in der Community nur noch im eigenen Saft?
Haben wir also lediglich die Ziele unserer Bemühungen eine oder mehrere Software-Generationen weitergedreht, sprich aus COBOL wurde Java, aus hierarchischen wurden relationale Datenbanken, aus Modularisierung wurden Container usw.
Werden uns neue Technologien wie KI oder Machine Learning die wirklichen Durchbrüche z.B. beim Reverse Engineering bringen?
Eine kritische Rückschau und offensive Vorschau erscheinen notwendig!

2 Teilnehmer

- **Prof. Dr. Jürgen Ebert**, 1999 einer der beiden Väter der WSR(E)-Reihe
- **Dr. Stefan Sauer**, als Vertreter des Arbeitskreis Langlebige Softwaresysteme (L2S2) mit seinen „Design for Future“-Workshops (DFF)
- **Harry Sneed**, der Reengineering-Pionier in Deutschland und weltweit renommiert
- **Jens Borchers**, fast durchgehend bei allen WSR(E) anwesend, Moderation der Diskussion
- …. und **das Auditorium**!

3 Wesentliche Ergebnisse

Zunächst wird festgestellt, dass die in [2] grundlegend dargestellten Ansätze zum Reverse und Reengineering weiterhin ihre Gültigkeit besitzen. Sie werden anhand des nachfolgenden, aus [3] stammenden Schaubilds, diskutiert.



Danach ist Reengineering heute ein wesentliches Teilgebiet des Software Engineering, das sich allerdings primär auf die technischen Komponenten der Softwareentwicklung konzentriert und andere Aspekte wie z,B. den Betrieb der Anwendungssysteme eher untergeordnet sieht. Es wird diskutiert, ob dieses ein Defizit darstellt, weil z.B. immer noch zu wenige umfassende statistische Daten über den gesamten Lebenszyklus eines Anwendungssystems erfasst werden und damit auch für moderne Ansätze wie z.B. Machine Learning kaum ausreichende Datenbasen vorhanden sind.
Nicht ganz erfüllt haben sich nach Meinung aller Diskussionsteilnehmer die ursprünglichen Ansätze und Hoffnungen im Bereich Reverse Engineering. Zwar gelingt es mittlerweile, zumindest die Grundstrukturen und wesentlichen Abhängigkeiten auch großer Anwendungssysteme auf Basis des Source Codes zu dokumentieren, aber die ursprüngliche Hoffnung des „Trichtermodells – oben

alte Software rein, unten automatisch nach aktuellen Designprinzipien und Programmiersprachen erzeugte Software raus“ nicht realisieren lassen.
Hier wird sich zeigen, ob Ansätze aus der „Künstlichen Intelligenz“, die ja derzeit in aller Munde ist und deren Produkte eines Tages auch selbst Ziel von Reengineering werden könnten (hier gibt es noch viele neue Themen auch zur Dokumentation), helfen werden.
Einig sind sich alle Beiträge in der Diskussion, dass der Bedarf für Reengineering auch in den nächsten 20 Jahren nicht abnehmen wird – hier wurden die Themen Cloud-Migration, neue Programmiersprachen (Java wird nicht die letzte große Mainstream-Sprache bleiben und COBOL und andere prozedurale Sprachen begleiten uns immer noch), neue Designkonzepte (auf SOA folgt Microservices) genannt.
Man kann also davon ausgehen, dass es den Workshop in Bad Honnef mit den in der Word Cloud dargestellten Themen auch die nächsten 20 Jahre wohl noch geben wird und darauf freuen wir uns alle!

Literatur

[1] F.L. Bauer et al, SOFTWARE ENGINEERING - Report on a conference sponsored by the NATO SCIENCE COMMITTEE, Garmisch, Germany, 7th to 11th October 1968, http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF, Zugriff am 25.04.2019

[2] E. Chikofsky and J.H. Cross, “Reverse Engineering and Design Recovery: A Taxonomy”, IEEE Software, January 1990, pp. 13-17.

[3] Ulrike Baumöl, Jens Borchers, Stefan Eicker, Knut Hildebrand, Rainhard Jung, Franz Lehner, “Einordnung und Terminologie des Reengineering“, Informatik-Spektrum, vol. 19, pp. 191-195, 1996.