

Quasar Analytics: Aktive Qualitätssteuerung in großen und verteilten Softwareprojekten

Alex Hofmann

Capgemini sd&m Research
Capgemini sd&m AG
Carl-Wery-Str. 42
D-81739 München
alexander.hofmann@capgemini-sdm.de

1 Steigende Anforderungen an die Softwareentwicklung

Hoher Termin- und Budgetdruck, kürzere Entwicklungszeiten bei gleichzeitig komplexer werdenden Softwaresystemen, die zudem oft noch verteilt entwickelt werden: Die Anforderungen an die Entwicklung individueller Softwarelösungen steigen. Während auf die Optimierung von Prozessen in Entwicklungsprojekten bereits großer Wert gelegt wird, vernachlässigen viele Unternehmen die Produktqualität. Dabei macht sich eine schlechte Software-Qualität schnell bemerkbar: 60 bis 70 Prozent der IT-Budgets werden derzeit in Wartung und Weiterentwicklung investiert. Kosten und Risiken können aber reduziert werden, wenn die Produktqualität bereits im Entwicklungsprozess gesteuert wird. Dafür liefert Quasar Analytics Methodik und Werkzeuge: Qualitätseigenschaften von Software lassen sich eindeutig planen und Sollabweichungen werden früh erkannt. Software-Dienstleister können so ihre Projekte im vorgegebenen Budget- und Terminrahmen abwickeln und die Hauptanforderung der IT-Entscheider erfüllen: schnell stabile Systeme zu liefern.

Ziel von Quasar Analytics ist es, die Qualität der Software während des gesamten Entwicklungsprozesses anhand definierter Kriterien zu steuern. Diese Vorgehensweise geht deutlich über die heute zumeist betriebene Prüfung von Codequalität (mittels Findbugs, Checkstyle, CPD, JavaNCSS usw.) hinaus. Die gesamthafte Planung und Steuerung von Software-Qualität wird heute nur selten betrieben – auch weil dafür geeignete Werkzeuge und Standardisierungen fehlen.

2 Quasar Analytics als Gesamtansatz

Den Kern der Methodik bildet das sogenannte Qualitätsmodell, mit dem die Qualität der Software anhand von ganz konkreten Eigenschaften und Kennzahlen bewertet wird. Hierzu werden Sensoren in die Entwicklungsumgebungen integriert, die permanent und bereits sehr früh im Entwicklungsprozess Werte für die definierten Kennzahlen auf der Ebene von Komponenten messen. Die Werte werden in ein Software-Cockpit eingespeist. Darüber hat der Software-Architekt jederzeit Zugriff auf den aktuellen Qualitätsstand. Sollabweichungen werden schnell erkannt. Die Kennzahlen des Qualitätsmodells – das sogenannte „Blutbild der Software“ – sind damit ein Frühwarnsystem für Qualitätsprobleme und ein aktives Steuerungsinstrument für anspruchsvolle Softwareprojekte.

Auf der Qualitätssteuerung über das Qualitätsmodell setzt die zweite Stufe auf: Quality-Gates – definierte Prüfpunkte vor kritischen Projektphasen, anhand derer die inhaltliche Reife einer Phase überprüft wird. Das erste Quality-Gate findet bereits vor Abgabe des Angebots statt und leitet zur Spezifikation der Software über. Ein weiteres Gate befindet sich im ersten Drittel der Spezifikationsphase, um wesentliche fachliche Entwurfsentscheidungen zu überprüfen. Vor einer Realisierung in der Breite befindet sich ein Gate, das die Konstruktionsentscheidungen noch einmal im Abgleich mit den Anforderungen des Kunden spiegelt und die Einhaltung von standardisierten Entwurfsprinzipien prüft. Die Prüfungen werden von externen, nicht am Projekt beteiligten Experten durchgeführt, um eine größtmögliche Objektivität zu gewährleisten. Im Rahmen dieser Qualitätssicherung werden die im Qualitätsmodell gewonnenen Kennzahlen noch einmal explizit anhand vorgegebener Sollwerte beurteilt und durch weitere Messungen ergänzt.

Die dritte Stufe – definierte Test- und Abnahmeverfahren auf Komponentenebene – ergänzt die Qualitätskontrolle. Hierzu gehören während der gesamten Entwicklungsphase funktionale und nichtfunktionale Testläufe, die beispielsweise die fachliche Korrektheit sowie Performance und Stabilität des Systems prüfen. Eine kontinuierliche Integration prüft die Zusammenarbeit der einzelnen Komponenten.

Literaturverzeichnis

- [BB+78] B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. McLeod, M. Merritt: Characteristics of Software Quality, North Holland, 1978.
- [BW84] V.R. Basili, D.M. Weiss: A Methodology for Collecting Valid Software Engineering Data, IEEE Transaction on Software Engineering, SE-10, No.6, S.728-738.
- [BEJ06] R. Buschermöhle, H. Eekhoff, B. Josko: Erfolgs- und Misserfolgskriterien bei der Durchführung von Hard- und Software-Entwicklungsprojekten in Deutschland, BIS Verlag der Carl von Ossietzky Universität Oldenburg, 2006.
- [Fe96] N. E. Fenton, S. L. Pfleeger: Software Metrics – A Rigorous and Practical Approach, International Thomson Publishing Company, second Edition, 1996.
- [Gr92] R. B. Grady: Practical software metrics for project management and process improvement, Prentice Hall, 1992