

Keynote: Continuous Software Engineering

Wilhelm Hasselbring¹

Continuous Software Engineering ist seit Jahrzehnten ein kontinuierlich behandeltes Thema in der Forschung und der Praxis des Software Engineering. Traditionell fokussiert die kontinuierliche Softwareentwicklung auf Reverse und Re-Engineering Aktivitäten zur Weiterentwicklung langlebiger Softwaresysteme.

In den letzten Jahren hat dieses Thema in der Praxis durch die Verfügbarkeit von leistungsfähigen Werkzeugen zur Automatisierung stark an Bedeutung gewonnen. Gleichzeitig ist eine Betonung automatisierter Qualitätssicherungsmaßnahmen zu beobachten. Continuous Integration und Continuous Delivery/Deployment sind dazu aktuelle Schlagworte.

Für die kontinuierliche Softwareentwicklung ist es nun auch wichtig neben der Konstruktion und Weiterentwicklung von Software auch schon in der Entwicklung den späteren Betrieb im Rechenzentrum oder in eingebetteten Systemen zu berücksichtigen. Die DevOps-Bewegung zielt darauf ab die Zusammenarbeit von Softwareentwicklung (Dev für Development) und Betrieb (Ops für Operations) zu optimieren und Reibungsverluste zu vermeiden.

In der (agilen) Softwareentwicklung ist es das Ziel, schnell viele Features bereitzustellen. Im Betrieb ist es das Ziel, stabile Dienste bereitzustellen - häufige Änderungen werden hier traditionell als unerwünscht angesehen. DevOps verfolgt nun den Ansatz viele, stabile Releases bereitzustellen. Die dazu erforderliche Qualitätssicherung und Effizienzsteigerung wird durch die Automatisierung von Entwicklungs- und Betriebsaufgaben erreicht.

Generell führt die kontinuierliche Integration von Qualitätssicherungsmaßnahmen zu einer kontinuierlich hohen Qualität und damit zu vielen stabilen Releases. Zur kontinuierlichen Überwachung der resultierenden Softwaredienste und auch der sogenannten Deployment-Pipelines muss möglichst viel automatisiert gemessen und überwacht werden (Monitoring).

Durch DevOps bekommt das Thema Softwarearchitektur auch in der agilen Softwareentwicklung eine stärkere Bedeutung und Würdigung. Für DevOps ist es sinnvoll präskriptive und deskriptive Architektur-Modelle zu kombinieren. Präskriptive Modelle kommen aus der Softwareentwicklung (Forward Engineering). Deskriptive

¹ KoSSE, Universität Kiel, Deutschland, hasselbring@email.uni-kiel.de

Modelle kommen aus der Beobachtung der im Betrieb befindlichen Softwaredienste (Reverse Engineering durch dynamische Analyse). In diesem Vortrag werde ich diskutieren, warum Softwarearchitektur ein zentrales Artefakt an der Schnittstelle zwischen Entwicklung und Betrieb ist. Speziell Microservice-Architekturen und dem kontinuierlichen Monitoring der resultierenden Systeme kommt hier eine besondere Rolle zu.