

## **5. Workshop „Design For Future – Langlebige Softwaresysteme“**

Stefan Sauer<sup>1</sup>, Benjamin Klatt<sup>2</sup>, Thomas Ruhroth<sup>3</sup>

<sup>1</sup>Universität Paderborn, s-lab – Software Quality Lab  
Zukunftsmeile 1, 33102 Paderborn  
sauer@s-lab.upb.de

<sup>2</sup>FZI Forschungszentrum Informatik an der Universität Karlsruhe  
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe  
klatt@fzi.de

<sup>3</sup>TU Dortmund, Fakultät für Informatik  
Otto-Hahn-Straße 14, 44221 Dortmund  
thomas.ruhroth@cs.tu-dortmund.de

<http://akl2s2.ipd.kit.edu/veranstaltungen/dff2013/>

Auch Software altert! Dieses Problem ist vor allem bei großen betrieblichen Informationssystemen unter dem Begriff Legacy bekannt und wird sich in Zukunft noch weiter verschärfen. Zum einen gewinnen eingebettete Systeme immer größere Bedeutung, in denen aufwändige Software in langlebigen technischen Geräten eingesetzt wird. Zum anderen macht die steigende Vernetzung von Systemen in großen Anwendungslandschaften die Situation zunehmend komplexer. Diese Probleme haben enorme ökonomische Bedeutung. Wissenschaft und Industrie sind gefordert, neue Methoden der Softwaretechnik zu entwickeln, um die erheblichen Investitionen in große Softwaresysteme zu schützen und massive Probleme durch steigende Software-Erosion zu verhindern.

Aktuelle Ansätze der Softwaretechnik, vor allem modellbasierte Entwicklungsmethoden, Lifecycle-Management, Softwarearchitektur, Requirements und Re-Engineering, können die Situation verbessern, wenn sie geeignet weiterentwickelt und angewandt werden.

Der Arbeitskreis „Langlebige Softwaresysteme“ (AK L2S2) der GI-Fachgruppen Architekturen und Software-Reengineering will Wissenschaftler und Praktiker zusammenzubringen, die an diesen Themenstellungen Interesse haben. Im 5. Workshop „Design For Future – Langlebige Softwaresysteme“ sollen die oben geschilderte Entwicklung, Erfahrungen hierzu sowie Lösungsansätze sowohl aus praktischer als auch aus wissenschaftlicher Sicht beleuchtet werden, um die verschiedenen Facetten und Herausforderungen der Software-Alterung zu beherrschen. Es sollen sowohl Lösungen als auch praktische Erfahrungen betrachtet und diskutiert werden, um die Entstehung neuer Legacy-Probleme und die Erosion von Software zu verhindern. Beiträge werden insbesondere zu den folgenden Themen erwartet: anpassungsfähige und zukunftssichere Software-Architekturen; Evolution und Co-Evolution von Modellen und Code; Verhinderung von Software-Erosion; Re-Engineering zum Erkennen und Beheben von Legacy-Problemen; Entwicklungsmethoden und Lifecycle-Management für langlebige Softwaresysteme; Qualitätsmanagement für langlebige Softwaresysteme; Fallstudien zu den vorgenannten Themen; Praxis- und Erfahrungsberichte zu den vorgenannten Themen.

