

## <workshop>praktiken\_des\_programmierens</workshop>

### <datum>

Freitag, 1.12. 2006  
von 10.00 bis 16.00 Uhr

### </datum>

### <ort>

Freie Universität Berlin  
Institut für Philosophie / Hermeneutik  
Seminarraum, Thielallee 43, 14195 Berlin

### </ort>

### <idee>

Gabriele Gramelsberger, Sebastian Jekutsch, Robert Schmidt

### </idee>

### <konzeption und organisation>

Robert Schmidt  
Sfb 447, Kulturen des Performativen, www.sfb-performativ.de

### </konzeption und organisation>

### <inhalt>

Programmieren, also das Entwerfen, Schreiben, Lesen, Kopieren, Anpassen, Verändern und Testen von Software ist eine Praktik, durch die über Branchen- und Disziplinengrenzen hinweg in den unterschiedlichsten arbeitsweltlichen und wissenschaftlichen Kontexten Wissen erzeugt und prozessiert sowie Realität konfiguriert wird. In den zuständigen Fachdisziplinen überwiegen Ansätze, die statt der Praktiken die Modelle, die Verfahrensvorschriften, die Resultate oder die Anforderungen an Software in den Mittelpunkt rücken. Die sozialen und epistemischen Praktiken des Programmierens, denen in der zeitgenössischen Wissensgesellschaft zentrale Bedeutung zukommt, werden demgegenüber bislang noch wenig berücksichtigt. Im interdisziplinär ausgerichteten Workshop stehen diese Praktiken im Mittelpunkt von Zugängen aus der Informatik, der Philosophie und der Soziologie. Den Ausgangspunkt bildet die Bemühung, Programmieren als Modellierungs-, Entwicklungs- und Wissensarbeit zu verstehen und zu beschreiben. In dieser Perspektive sollen verschiedene arbeitsweltliche und wissenschaftliche Kontexte des Programmierens miteinander verglichen und zueinander in Beziehung gebracht werden. Folgende Frage- und Forschungsrichtungen werden dabei anvisiert:

Die Perspektive auf Praktiken des Programmierens ist in empirischer Hinsicht mit einer Reihe von methodischen Problemen verbunden: Unter welchen Voraussetzungen ist Programmieren v. a. hinsichtlich der mentalen Praktikenbestandteile wie Nachdenken, Schlussfolgern etc. beobachtbar? Welcher Detaillierungsgrad solcher Beobachtungen ist sinnvoll? Welche Methoden und Tools stehen zur Verfügung, um die Interaktion zwischen Programmierer und Code zu dokumentieren und zu interpretieren? Wie können solche Analysen zur Fehlerreduktion und zur Verbesserung des Herstellungsprozesses beitragen?

Im Bereich der Softwareentwicklung sind in jüngerer Zeit sogenannte agile Ansätze (z.B. Extreme Programming) entstanden, die in Abkehr von traditionellen ingenieurmäßigen Modellen, in denen zwischen Entwurf und Codierung strikt getrennt wird, das Programmieren zur Schlüsselaktivität der Softwareentwicklung erklären. Wie antworten solche Ansätze auf die praktischen Probleme der Softwareentwicklung? Wie vollzieht sich die Umsetzung solcher Modelle in realen Umgebungen und welche Probleme entstehen dabei?

Computerprogramme sind immer Modelle von etwas, das durch diese Modelle zugleich erzeugt und verändert wird. Wie wird in den Praktiken des Programmierens Wissen produziert? Wie sind die Beziehungen zwischen geschriebenem Code und dessen Referenz, also dem, was durch den Code (re-)präsentiert, modelliert und auf-/ ausgeführt wird? Welchen epistemischen (performativen, theoretischen, narrativen, intertextuellen etc.) Status hat der in den Praktiken entstehende Code? Wie sind die Beziehungen zwischen der Komplexität des Codes und der Komplexität der epistemischen Objekte (also dessen, was codiert wird)?

Als paradigmatische Wissensarbeit existieren die Praktiken des Programmierens auch als öffentliche, mediale und populärkulturelle Erscheinungsformen. An welche Aspekte der Praktiken des Programmierens knüpfen solche symbolischen Repräsentationen an? Artikuliert und zeigt sich Programmieren in den Praktiken als Ingenieurstätigkeit, als wissenschaftliche oder als künstlerische Arbeit? Wie entstehen in den Praktiken des Programmierens die sozialen und kulturellen Identitäten, die Berufsbilder dieser Tätigkeiten?

### </inhalt>

## <workshop> praktiken\_des\_programmierens </workshop>

<inhalt>siehe\_oben  </inhalt>

<programm>siehe\_unten  </programm>

<plakat>siehe\_rückseite  </plakat>

<datum>Freitag, 1. 12. 2006 von 10.00 bis 16.00 Uhr</datum>

<ort>FU Berlin, Institut für Philosophie / Hermeneutik  
Seminarraum, Thielallee 43, 14195 Berlin</ort>

### <programm>

10.00 - 10.15 Uhr:  
Einführung (Sebastian Jekutsch, Robert Schmidt)

10.15 - 11.00 Uhr:  
Sebastian Jekutsch (Informatik, FU Berlin):  
Der Mikroprozess des Programmierens

11.00 - 11.45 Uhr:  
Lutz Prechelt (Informatik, FU Berlin): Extreme Programming:  
Selbstorganisation durch gezielt ineinandergreifende Praktiken

11.45 - 12.00 Uhr:  
Kaffeepause

12.00 - 12.45 Uhr:  
Gabriele Gramelsberger (Philosophie, FU Berlin):  
Praktiken des wissenschaftlichen Programmierens

12.45 - 14.00 Uhr:  
Mittagspause

14.00 - 14.45 Uhr:  
Gunter Gebauer (Philosophie, FU Berlin):  
Wittgenstein über Regelfolgen

14.45 - 15.30 Uhr:  
Robert Schmidt (Soziologie, FU Berlin):  
Der innere Raum des Codes. Programmieren als soziale Praktik.

15.30 - 16.00 Uhr:  
Abschlussdiskussion

### </programm>

<anmeldungen bitte an>  
schmiro@zedat.fu-berlin.de  
</anmeldungen bitte an>

