ABSTRACT
Understanding software engineering issues for autonomic computing systems is critical for the software and information technology sectors, which are continually challenged to reduce the complexity of their systems. To be autonomic, a system must know itself as well as its boundaries and its environment, configure and reconfigure itself, continually optimize itself, recover or heal from malfunction, protect itself, and function in a heterogeneous world—while keeping its complexity hidden from the user. The goal of this workshop is to bring together researchers and practitioners, who investigate concepts, methodologies, techniques, technologies, and tools to design and evolve autonomic software.

Categories & Subject Descriptors: D.2.10 [Software Design]: D.2.11 [Software Architectures]
General Terms: Design
Keywords: Autonomic computing, self-managed systems

1. MOTIVATION AND GOALS
While there are several workshops [1, 4, 5] that deal with autonomic computing systems, there are few workshops that focus on software engineering issues, i.e., how do we design, build, and evolve such software systems so that they can meet given—and evolving—requirements for particular classes of users and/or applications. Most existing systems cannot be re-designed and re-developed from scratch to incorporate autonomic capabilities. Rather self-management capabilities have to be added gradually and incrementally—one aspect at a time. The references include some excellent starting points [2, 3, 6]. With the proliferation of autonomic applications, users will impose ever-more demands with respect to functional and non-functional requirements for autonomy.

The goal of this workshop is to exchange opinions, advance ideas, and discuss preliminary results among researchers and practitioners who investigate concepts, methodologies, and tools to design and evolve autonomic software. The topic of self-managed systems has been studied in a large number of specific areas, including databases, robotics, control systems, fault-tolerant computing, agents, adaptive systems, neural networks, and others. In this workshop however, we will concentrate on the design and evolution of autonomic application software.

2. TOPICS OF INTEREST
Topics of interest include, but are not limited to, architectural styles, attribute-based architectural styles, and architecture patterns for autonomic elements and systems, designing high-variability software, designing self-managed systems, evolving autonomic software, injecting autonomicity into legacy systems, integration mechanisms, methods for evaluating complex tradeoffs, adoption of autonomic systems, or assessing the user experience in self-managed systems.

Applications of interest include, but are not limited to, web services, applications involving software that helps people with special needs live their lives, software that integrates multiple heterogeneous components, such as an inter-organizational workflow system that coordinates production or service processes, or autonomic systems serving the information economy.

3. REFERENCES