Current Best Practices in Software Architecture

Dr. Paul C. Clements
Software Engineering Institute / Carnegie Mellon University, USA

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.

Software architecture represents the most critical engineering opportunity for achieving important quality attributes in complex software systems: performance, availability, platform independence, security, etc. The software architecture for a system embodies the earliest and most far-reaching design decisions about the software, which makes it very important to design correctly. A system with the wrong software architecture will fail.

This seminar series will present a series of talks on the best current practices in creating and using software architecture. Each session will last 3 hours; some sessions may include group exercises. The tentative schedule is as follows:

	Topic
1	Overview of software architecture: What is it? Why is it important? What is it used for?
2	Understanding architectural requirements: Quality attributes, quality attribute scenarios, quality requirements elicitation and capture
3	Designing software architectures: Patterns, styles, and tactics. Views. A method for designing architectures.
4	Documenting software architectures: How do we write down an architecture so that others can use it?
5	Evaluating software architectures: How do we know that our architecture is the right one?
6	Architectures for software product lines: What is a software product line? Why are they important? What role does architecture play?

"Software Architecture in Practice" (1998, second edition 2003)
"Evaluating Software Architectures: Methods and Case Studies" (2001)
"Documenting Software Architectures: View and Beyond" (2002).