

Incorporating Software Architecture in the Computing Curriculum

Lawrence G. Jones
*Research, Technology and Systems Solutions
Program
Software Engineering Institute
Carnegie Mellon University
lgj@sei.cmu.edu*

Martin L. Barrett
*Department of Computer and Information
Sciences
East Tennessee State University
barrettm@etsu.edu*

Ayse Bener
*Dept of Computer Engineering
Bogaziçi University, Istanbul
bener@boun.edu.tr*

Steve Chenoweth
*Department of Computer Science
Rose-Hulman Institute of Technology
chenowet@rose-hulman.edu*

Amine Chigani
*Software Engineering Research Group
Department of Computer Science
Virginia Tech
achigani@vt.edu*

Abstract

This workshop introduces the concepts of software architecture and how to incorporate these concepts into the computer science and software engineering curriculum. Participants will learn techniques used in industry to specify quality attributes critical to system performance, modifiability, and availability, and to use those attributes to drive the system structure using well-known architectural styles. Exercises will be used to demonstrate the techniques and to practice effective methods for students to learn the techniques in CS and SE classes. Participants will become part of the community of educators sharing educational resources in software architecture.

1. Intended audience

Undergraduate and graduate computer science and software engineering instructors who want to understand the importance of software architecture in the creation of software systems and wish to learn how to incorporate software architecture concepts into the curriculum. Software architecture education prepares students to work on large software systems. Software architecture represents the earliest design decisions that are both the most difficult to get right and the hardest to change downstream. System qualities, such as performance, security, availability, interoperability, or modifiability, are largely supported or precluded once the software architecture has been established.

2. Materials provided

- (1) Paper and electronic copies of workshop notes.
- (2) Templates for producing software architecture work products.
- (3) Answers to exercises used in the workshop.

3. Rough Agenda

1. Overview of software architecture. *15 minutes.*
2. Quality attributes and their role in architecture. *30 minutes.*
 - overview of quality attribute categories
 - scenario examples
 - brainstorming by participants of additional examples and uses (in small groups)
3. Eliciting quality attributes. *30 minutes.*
 - exercise in elicitation and ranking quality attributes (in small groups)
4. Architectural style guide example. *30 minutes.*
 - overview of architectural viewtypes and styles
 - component-and-connector example
 - module example
 - allocation example
 - exercise in using multiple views for architecture analysis
5. An example of using quality attributes to create a software architecture. *30 minutes.*
 - overview of tactics
 - example of performance tactics
 - example of modifiability tactics
 - exercise in using availability tactics
6. Software architecture in the curriculum. *30 minutes.*
 - undergraduate software engineering courses
 - dedicated software architecture courses
 - examples of successful learning methods in software architecture
 - brainstorming of additional ideas (in small groups)

4. Audio/Visual and Computer requirements

A digital projector for presenters, a flipchart with pens, whiteboard with markers.

5. Space and Enrollment restrictions

No special restrictions. We intend to form participants into groups of 4-5 people during parts of the workshop, so it would help if the space provided allowed for shifting of chairs to enable this.

6. Other critical information

The software architecture material used here is based on workshops developed by the Software Engineering Institute's Research Technology and Systems Solutions Program.