

WVU LDC SEE CS 430

Fall 2004

Copyright 1999, 2001, 2002
R. S. Pressman & Associates, Inc.

For University Use Only

May be reproduced **ONLY** for student use at the university level
when used in conjunction with **Software Engineering: A Practitioner's Approach**.
Any other reproduction or use is expressly prohibited.

These courseware materials are distributed in conjunction with **Software Engineering: A Practitioner's Approach**. All rights are reserved
with permission by R. S. Pressman & Associates, Inc., copyright © 1999, 2001, 2002.

1

Why Architecture?

The architecture is not the operational software. Rather, it is a representation that enables a software engineer to:

- (1) analyze the effectiveness of the design in meeting its stated requirements,
- (2) consider architectural alternatives at a stage when making design changes is still relatively easy, and
- (3) reduce the risks associated with the construction of the software.

These courseware materials are distributed in conjunction with **Software Engineering: A Practitioner's Approach**. All rights are reserved
with permission by R. S. Pressman & Associates, Inc., copyright © 1999, 2001, 2002.

2

Why is Architecture Important?

- Representations of software architecture are an enabler for communication between all parties (stakeholders) interested in the development of a computer-based system.
- The architecture highlights early design decisions that will have a profound impact on all software engineering work that follows and, as important, on the ultimate success of the system as an operational entity.
- Architecture "constitutes a relatively small, intellectually graspable model of how the system is structured and how its components work together" [BAS03].

These courseware materials are distributed in conjunction with **Software Engineering: A Practitioner's Approach**. All rights are reserved
with permission by R. S. Pressman & Associates, Inc., copyright © 1999, 2001, 2002.

3

Data Design—Component Level

1. The systematic analysis principles applied to function and behavior should also be applied to data.
2. All data structures and the operations to be performed on each should be identified.
3. A **data dictionary** should be established and used to define both data and program design.
4. Low level data design decisions should be deferred until late in the design process.
5. The representation of data structure should be known only to those modules that must make direct use of the data contained within the structure.
6. A library of useful data structures and the operations that may be applied to them should be developed.
7. A software design and programming language should support the specification and realization of abstract data types.

These courseware materials are distributed in conjunction with **Software Engineering 4: Software's Agenda**. All content is provided with permission by R.E. Proctor and Associates, Inc., copyright © 1999, 2001, 2002.

6

Architectural Styles

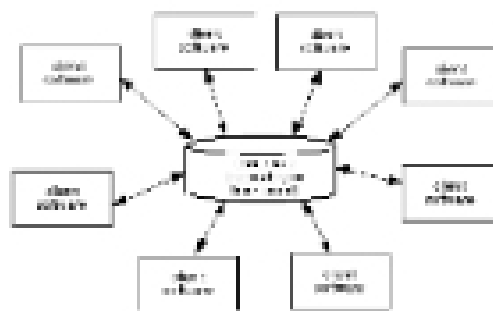
Each style describes a system category that encompasses: (1) a set of components (e.g., a database, computational modules) that perform a function required by a system, (2) a set of connectors that enable "communication, coordination and cooperation" among components, (3) constraints that define how components can be integrated to form the system, and (4) semantic models that enable a designer to understand the overall properties of a system by analyzing the known properties of its constituent parts.

- Data-centered architectures
- Data flow architectures
- Call and return architectures
- Object-oriented architectures
- Layered architectures

These courseware materials are distributed in conjunction with **Software Engineering 4: Software's Agenda**. All content is provided with permission by R.E. Proctor and Associates, Inc., copyright © 1999, 2001, 2002.

7

Data-Centered Architecture



These courseware materials are distributed in conjunction with **Software Engineering 4: Software's Agenda**. All content is provided with permission by R.E. Proctor and Associates, Inc., copyright © 1999, 2001, 2002.

8

