Garlan and Shaw describe several architectural styles for software:

- Pipes and Filters
- Object-Oriented Organization
- Event-based, Implicit Invocation
- Layered Systems
- Repositories
- Table Driven Interpreters
Choose three.

Give an example of each from software that you use or know well.

*Work in groups of 2 or 3.*
How easy is it to determine the architecture of a system from its user interface?

Is it possible that a system could be implemented with different architectures, and as a result behave differently?
One of the driving goals of a software engineering discipline: a handbook for design and implementation.
Even scientists have such handbooks, albeit at a lower level.
We need a way to **reliably** and **repeatably** map problem types onto solution types.

Recur “a million times, but never the same way”.
A first effort, documenting work distributed across many disciplines in a single whole. 1996!
Choose one:
- Pipes and Filters
- Event-based
- Layered Systems
- Repositories

Describe how we could build Arnold’s video rental software using this architecture.

Work in your same groups.
<discuss candidate designs>

Is an architecture ever not possible for a given problem? What characteristics make one architecture more suitable than another for a given problem?

Are there any questions about these architectural styles?
Which architectural style is best?

... for a given problem
... for a given technology
... ...
Tell, Don’t Ask

What does it mean?
Is it a good idea, or hokum?
The Law of Demeter

Is it a law or a good suggestion? How can we know when it is okay to break the rule?
Designs are not good or bad so much as better or worse.

... in a given context.

When the context changes, the evaluation of quality changes. (Perhaps we should think in terms of suitability or fitness.)

Keep this in mind whenever you encounter any program, design, or system.

Or an architectural style.