Elements of a successful software architecture course

David Garlan
Len Bass and Eric Dashofy

ASEE&T 2010 ▪▪ March 12, 2010

Proposed Agenda

- What are the core ideas of software architecture that should be taught to students?
- How is "architectural thinking" different from programming?
- What is a good architecture assignment?
- How does what you teach depend on the background of the students in the class?
- Should we be teaching software architecture at the undergraduate level?
- What should a *second* course in software architecture cover?
- What can you teach in a 3-day intensive course, for example, to industry?

What are the core ideas of software architecture that should be taught to students?

- One point of reference is the CMU Architectures for Software System
  - Taught since the early 1990s at CMU in the Master in Software Engineering Course
  - Originated as an experimental course, but later recognized as a core component of professional SE education
  - Has undergone many changes over the years
  - But some things have remained constant

What we teach now -1

- General Concepts
  - What is software architecture
  - Basic concepts: views, styles, patterns
- Principles of Architecting
  - Understanding architectural requirements
  - Architecture styles and tactics
  - Product lines and integration frameworks
  - From architecture to code

What we teach now -2

- Architecture in Practice
  - Evaluating architectural designs
  - Handling architectural problems
  - Documenting a software architecture
  - Presenting an architecture to others
  - Architecture for X (security, usability, reliability, etc.)

What we teach now

- Key concepts
  - Software architecture is distinct from programming.
  - Styles are a fundamental building block for architectural practices.
  - Understanding requirements is essential.
  - It is important to understand the link between architecture and code.
  - Notations matter. Programming languages do not matter.
  - Documentation is hard to do right, but worth the effort.
  - Quality attributes determine the success of an architecture (reliability, performance, etc.)

What we no longer teach

- Module interconnection languages
- History of software engineering
- Problem frames

What are the core ideas of software architecture that should be taught to students? *(continued)*

How is "architectural thinking" different from programming?

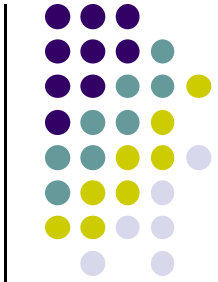
- Architectural thinking is about understanding engineering tradeoffs.
- Architectural thinking is about thinking at a systems level.
- Architectural thinking requires a comprehensive understanding of multiple stakeholders' concerns

How is "architectural thinking" different from programming? *(continued)*

What is a good architectural assignment?

- Worth distinguishing between different kinds of assignments
  - Short single-class assignments
  - 2-week team-based architecture assignments
  - Projects
- We use all three

The Context*

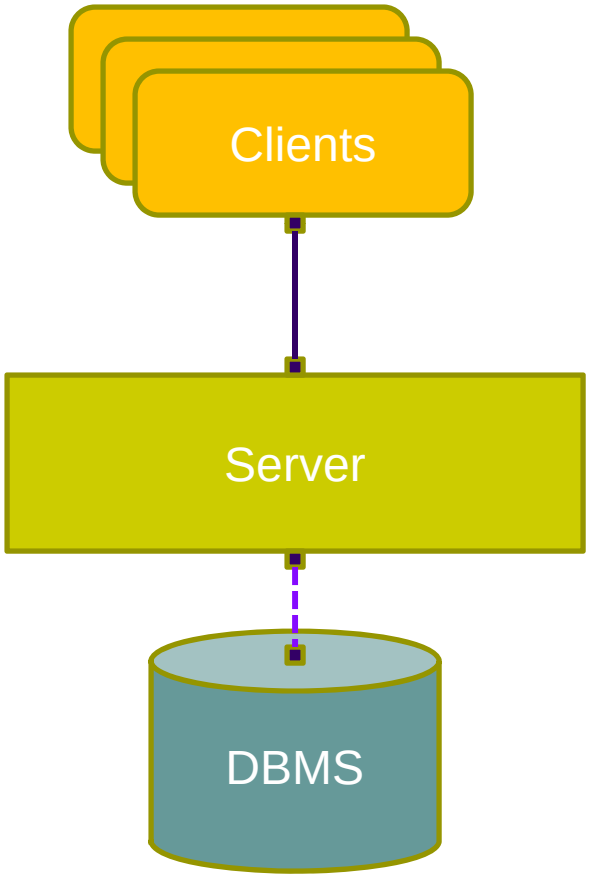


- Today you are a consultant for an organization that wants to create a web application that:
  1. Gets some data from the user
  2. Checks for validity and does some computation
  3. Stores the result so it can be accessed later
- Key functionality:
  - Must maintain a repository of user info
  - Users can update their mailing address, SSN, etc.
  - Validation involves checking things like zip code has right number of digits; zip code is consistent with the listed city, etc.
  - May be incorporated into a larger system at a later date.

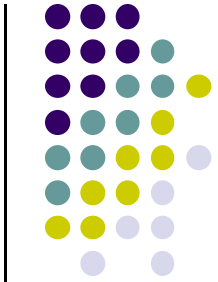
*Thanks to Ciera Jaspin

Given “Architecture”

Any issues with this representation?

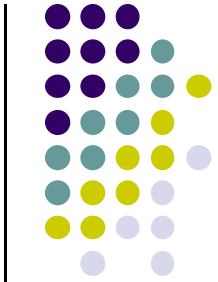


The Problem



- A key concern and question from the client is where should validation be performed?
  - On the Client?
  - On the Server?
  - In the Database itself?
  - Other?

Your task



- How would you advise your customers with regard to this system?
  - What are the important issues that your customer should be concerned with?
    - quality attributes, business concerns, technology, others…
  - How do those issues affect the architectural choice(s) for where validation is performed?
- Produce a couple of slides that explain your team's point of view and the advice you would give your customer.

Summary

- Summary point 1 (perhaps by David only)
- Summary point 2 (perhaps by David only)
- …

Undergraduate Software Architecture Course?

- Do undergraduates have sufficient maturity?
- What can you *not* do with undergraduates?

A second course in software architecture?

- What can one do at an advanced architecture level?
- Does this necessarily involve domain specialization?

What can you do for industrial training?

- What can you hope to accomplish in a 3-day course?
- How can one best work with industry to train their engineers?
- Examples: Sony, Samsung, Xerox