Style Intro &
Early Feedback Evaluation

Reid Holmes
Architectural styles

- Some design choices are better than others
  - Experience can guide us towards beneficial sets of choices (patterns) that have positive properties
- An architectural style is a named collection of architectural design decisions that:
  - Are applicable to a given context
  - Constrain design decisions
  - Elicit beneficial qualities in resulting systems
Architectural styles

A set of architectural design decisions that are applicable to a recurring design problem, and parameterized to account for different software development contexts in which that problem appears.

e.g., Three-tier architectural pattern:
Architectural styles

- Defines a family of architectures that are constrained by:
  - Component/connector vocabulary
  - Topology
  - Semantic constraints

- When describing styles diagrammatically:
  - Nodes == components (e.g., procedures, modules, processes, databases, …)
  - Edges == connectors (e.g., procedure calls, events, db queries, pipes, …)
Good properties of an architecture

- Result in a consistent set of principled techniques
- Resilient in the face of (inevitable) changes
- Source of guidance through product lifetime
- Reuse of established engineering knowledge
“Pure” architectural styles

- Pure architectural styles are rarely used in practice.

- Systems in practice:
  - Regularly deviate from pure styles.
  - Typically feature many architectural styles.

- Architects must understand the “pure” styles to understand the strength and weaknesses of the style as well as the consequences of deviating from the style.
IN CLASS
Outline

- Feedback on early evaluation forms
- Map activities back to intended learning outcomes

Critique an existing architecture or design.
Differentiate how various architectural styles and design patterns enhance and degrade a system’s functional-and non-functional properties.
Generate and justify architecture and/or design given a collection of requirements.
Produce and present concise and unambiguous architecture and design descriptions.
Create and implement an architecture and design, refining it into a complete system.
Early Evaluation Feedback

- What will be on the final exam?
  - Any content posted to the course web page
    - Slides [easy questions]
    - Videos [easy questions]
  - Any content covered in class [hard questions]
  - The projects will not be evaluated (directly)
Early Evaluation Feedback

- Recap videos in class.
- Why not more examples in the videos?
- Video quality / takes / editing concerns.
- Course feels too app centric
- Why multiple platforms?
Architectural Analogy

- Kitchen design activity.
- What are the architectural components?
  - How are they related to each other?
  - What connectors exist?
  - Why did you choose the components / connectors / topology you did?
- How do the connectors bind the components?
- Why is software arch. like traditional arch.?
- Why is software arch. not like traditional arch.?
Architectural Decomposition

- Generate an architecture for an *automated shopping cart*.
  - Identify the key components and connectors.
  - Derive a system topology.
  - Justify your decomposition.
    - Why these components?
    - Does the architecture adequately capture the broad system goals?
    - What are the strengths and weaknesses of the proposed architecture?
Architectural Tradeoffs

- Generate an architecture for a *context-aware notification system*.
  - Identify NFPs for a given stakeholder.
    - Justify why those NFPs matter.
  - Determine how those NFPs influence the architecture of the system.
  - Compare the architectures derived when different stakeholders care about divergent NFPs.
Completeness & Consistency

- The Spec is Right.
  - For a given system description, can we identify:
    - Aspects that are inconsistent
    - Aspects that are incomplete
  - How can we build a description that all stakeholders can understand and reason about?
  - What is the right level of abstraction for an architectural document?
  - What tools and techniques can help us generate complete and consistent system descriptions?